Asian Journal of Electrical and Electronic Engineering

Volume 4 Issue 2 September 2024

eISSN 2785-8189

9 772785 818002

https://alambiblio.com/ojs/index.php/ajoeee

AIMS & SCOPE OF THE ASIAN JOURNAL OF ELECTRICAL AND ELECTRONIC ENGINEERING

The **Asian Journal of Electrical and Electronic Engineering** (**AJoEEE**), published biannually (March and September), is a peer-reviewed open-access journal of the **AlamBiblio Press**.

The Asian Journal of Electrical and Electronic Engineering publishes original research findings as regular papers and review papers (by invitation). The Journal provides a platform for Engineers, Researchers, Academicians, and Practitioners who are highly motivated to contribute to the Electrical and Electronics Engineering disciplines. It also welcomes contributions that address the developing world's specific challenges and address science and technology issues from a multidisciplinary perspective.

REFEREES' NETWORK

All papers submitted to AJoEEE Journal will be subjected to a rigorous reviewing process through a worldwide network of specialized and competent referees. Each accepted paper should have at least two positive referees' assessments.

SUBMISSION OF A MANUSCRIPT

A manuscript should be submitted online to the Asian Journal of Electrical and Electronic Engineering (AJoEEE) website https://journals.alambiblio.com/ojs/index.php/ajoeee/. Further correspondence on the status of the paper could be done through the journal's website

COPYRIGHT NOTICE

ASIAN JOURNAL OF ELECTRICAL AND ELECTRONIC ENGINEERING

eISSN 2785-8189

9 7 7 2 7 8 5 8 1 8 0 0 2

Published by:

AlamBiblio Press,
PV8 platinum Hill
Jalan Melati Utama, 53100 Kuala Lumpur, Malaysia
Phone (+603) 2713 7308

Whilst the publisher and editorial board make every effort to see that no inaccurate or misleading data, opinion or statement appears in this Journal, they wish to make it clear that the data and opinions appearing in the articles and advertisements herein are the responsibility of the contributor or advertiser concerned. Accordingly, the publisher and the editorial committee accept no liability whatsoever for the consequence of any such inaccurate or misleading data, opinion or statement.

ASIAN JOURNAL OF ELECTRICAL AND ELECTRONIC ENGINEERING

Volume 4, Issue 2, September 2024

TABLE OF CONTENTS

ASIAN JOURNAL OF ELECTRICAL AND ELECTRONIC ENGINEERING
Vol. 4 No. 2 2024
E-ISSN: 2785-8189

Improving the Privacy in Wireless-Enabled 5G Networks: A Lightweight Protocol for IIoT Communications

Mamoon M. Saeed[1*], Rashid A. Saeed[2], Mohammed Suliman Elbashier[2], Elmustafa Sayed Ali[3], and Zeinab E. Ahmed[2]

[1]*Department of Communications and Electronics Engineering, Faculty of Engineering, University of Modern Sciences (UMS), Yemen*

[2] *College of Electronics Engineering, Faculty of Engineering, Sudan University of Science and Technology, Sudan*

[3] *Department of Electrical & Electronic Engineering, Faculty of Engineering, Red Sea University, Sudan*

*Corresponding author: mamoon530@gmail.com

(Received:7 August 2024; Accepted: 27 August 2024)

Abstract— The vision and major elements of the fifth generation (5G) ecosystem have previously been explored. We examine how security may impact the envisioned 5G wireless systems the challenges and potential solutions to aid in these efforts and define the security and privacy aspects of 5G networks. 5G networks have provided solutions for quicker machine control, problem identification, performance analysis, and data access. Interaction between Internet of Things (IoT) nodes occurs across an unsecured wireless channel, which has positive and negative effects. Despite being physically separated, unauthorized nodes could communicate via an unprotected wireless channel to gather data and take over industrial devices. Secure sessions can mitigate these risks, but it might be challenging to construct a secure session over a weak channel. To address this issue, the Variable Identification (VID) is used. VID offers a simple key exchange platform to authorized Industry Internet of Things (IIoT) nodes while guarding against unauthorized use. The lightweight changeable pseudonyms used by VID for trust-building are selected at random from a pool discovered in the home network and terminal devices. All IDs are chosen at random from a pool and are used to protect data against forgery, replay, alteration, impersonation, and man-in-the-middle attacks, among other things, between the home network and terminal equipment. The ProVerif tool is used to evaluate the suggested system, and the findings demonstrate that it is trustworthy and resistant to prospective attacks.

Keywords: *Wireless, Privacy, IIoT, Security threats, 5G networks*

1. INTRODUCTION

A new era of connectivity and automation in the industrial sector has been brought about by the quick development of 5G networks and the broad uptake of the Industrial Internet of Things (IIoT). However, serious privacy and security problems are raised by the widespread usage of wireless communication in IIoT systems [1, 2]. In wirelessly equipped 5G network environments, there is an increased danger of illegal access, data breaches, and privacy leaks due to continuous connectivity and data sharing across devices. Thus, to protect sensitive data in IIoT communication, it is imperative to design strong privacy-enhancing methods, especially lightweight protocols [3].

DOI: https://doi.org/10.69955/ajoeee.2024.v4i2.62

5G networks, which are wirelessly enabled, present a variety of privacy challenges. The widespread collection of data and the extensive use of sensors and actuators increase the risk of privacy violations. Furthermore, communicating via wireless creates weaknesses that bad actors can take advantage of. Innovative approaches that balance resource limitations, energy efficiency, and privacy protection are needed to meet these problems [4].

In 5G network environments, privacy risks have been reduced by utilizing established privacy-enhancing strategies such as access control methods, authentication protocols, encryption algorithms, and anonymization techniques [5]. However, these methods frequently come with a high processing cost, connection latency, and scalability issues, which makes them less appropriate for IIoT devices with limited resources. It is therefore essential to build lightweight protocols, especially for IIoT communication.

To improve privacy in wirelessly enabled 5G networks, this literature review will examine the state of research and developments in this area, with an emphasis on the creation of lightweight protocols for IIoT communication. The review will look into the privacy issues that wireless-enabled IIoT systems present, examine current privacy-enhancing strategies, and assess how well they work to solve privacy issues. It will also explore the developments in lightweight protocols that are suited to the particular needs of IIoT communication, taking into account things like resource optimization, energy efficiency, and privacy preservation [6 – 8].

This study aims to shed light on the state of privacy in wirelessly enabled 5G networks by undertaking an extensive literature review. It seeks to highlight new trends and technologies, point out the advantages and disadvantages of current methods, and suggest possible directions for further study and invention [9]. To ensure privacy and security in the wirelessly connected 5G networks era, the ultimate goal is to encourage the development of effective and privacy-preserving protocols that can be effortlessly integrated into IIoT communication [10, 11].

The remainder of this work is arranged in the following manner. In Sect. 2, network security architecture in mobile wireless is discussed. Sect. 3 presents privacy-preserving in mobile wireless, followed by Sects. 4 and 5 discussions of the related works system model, and adversary model., Sects. 6 and 7 discuss the proposed scheme and analyze the key features of the proposed solution. Finally, Sect 8 presents the conclusion.

2. NETWORK SECURITY ARCHITECTURE IN MOBILE WIRELESS

Mobile networks have relied on the physical storage of symmetric keys in a subscriber identity module, also known as a subscriber identity module (SIM) card, since the beginning of digital mobile communication in 2G. Additional cryptographic procedures for mutual authentication were implemented, and encryption algorithms shifted from customary to international standards. However, the security approach of 5G is still heavily reliant on SIM cards [12]. Even though SIM cards have shrunk in size (to the "micro" size), they still need to be inserted into devices, limiting their applicability to IoT. The development of eSIMs somewhat addresses this issue, although physical size issues remain. iSIMs, which are now in development, could be used in future devices as part of the System-on-chip concept, while operators are opposed owing to the potential loss of control [13].

2.1 The requirements for a wireless network security architecture model

Traditional SIM cards use tried-and-true symmetric key encryption that has grown to billions of users. However, it has flaws with IoT, privacy, network authentication, and bogus base stations. One important topic is whether symmetric cryptography will give way to asymmetric public/private keys. This has never been done on such a large scale before. 5G aims to provide authentication via a public-key infrastructure in addition to SIM (PKI) [14]. The core of 5G will be a collection of microservices that communicate over HTTPS. Transport Layer Security (TLS) uses elliptic curve cryptography (ECC) to enable authentication, confidentiality, and integrity for such communication. However, this has not yet been implemented and can be put off until 6G [15].

This section answers some of the most frequently asked questions about the 6G security concept. Will physical SIM cards still be used in devices? Will the majority of IoT devices have software SIM clones or Trusted Platform Modules? Although certificate revocation and Certificate Authority (CA) break-ins are possible, a certificate system for the WWW works. The Domain Name Scheme Security Extensions (DNSSEC) is an example of an asymmetric key system being gradually deployed. A critical prerequisite for asymmetric encryption is the prevention of man-in-the-meddle attacks. Using an IPsec Virtual Private Network (VPN) can enable rudimentary isolation of user traffic; however, the more advanced use of network slicing techniques in 6G will be an open research subject, as it may expose the network to new vulnerabilities [16].

2.2 *Evolution of Mobile Security*

Cloning, unlawful physical attacks, eavesdropping, encryption issues, authentication and authorization problems, and privacy issues plagued the first generations of mobile networks (i.e., 1G, 2G, 3G) [7]. Then, the security threat landscape transformed with increasingly advanced attack scenarios and powerful attackers. Fig. 1 depicts the progression of the telecommunication network security landscape from 4G to the envisioned 6G future. The execution of wireless applications posed a security and privacy danger to 4G networks. Media access control (MAC) layer security threats (e.g., denial of service (DoS) attacks, eavesdropping, and replay attacks) and malware applications are common examples (e.g., viruses, tampering with hardware).

Security and privacy risks pose problems in 5G access, backhaul, and core networks [18]. The most prevalent security challenges in 5G are cyberwar and critical infrastructure threats, Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) related threats, and cloud computing-associated threats [19 - 22]. SDN can pose a security risk in several ways, including exposing important Application Programming Interfaces (APIs) to unwanted software, introducing Open Flow, and centralizing network control (i.e., making it vulnerable to DoS attacks) [23]. Above all, the increased linked intelligence in telecommunication networks and sophisticated networking and AI/ML technologies are the most crucial driving forces in the 6G vision. However, in many circumstances, the alliance between AI and 6G could be a double-edged sword when it comes to defending or infringing on security and privacy [24 - 27].

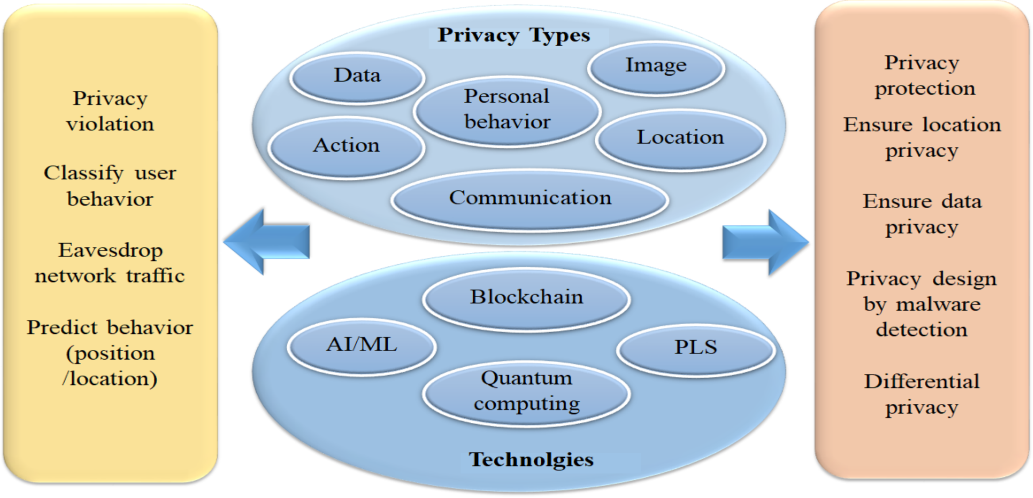


Fig. 1. Landscape of Privacy in Mobile Network.

3. PRIVACY-PRESERVING IN MOBILE WIRELESS

As 5G networks mature, AI-enabled smart applications are projected to become more prevalent, necessitating situational, context-aware, and personalized privacy solutions. Due to a wide and complicated set of unexpected privacy issues, traditional privacy-preserving techniques may not be well suited for future wireless applications [28 – 30]. Distributed ledger technologies, such as blockchain, may make it possible to deploy trustless computing between stakeholders while also providing mechanisms for network privacy protection. Among the security and privacy advantages of blockchain are immutability, transparency, verifiability, anonymity, and pseudonymity.

Blockchain can provide privacy-preserving data-sharing mechanisms, improve access control, provide key characteristics such as data integrity, traceability, and monitoring, and ensure efficient accountability mechanisms, among other things, and is seen as a viable option in Machine Type Communications in 6G [31].

When it comes to tackling important difficulties that are likely to develop in future intelligent 6G wireless applications, differential privacy (DP) approaches appear to be promising. Before sending the final output to the allocated server, DP perturbs the actual data using artificial design random noise functions [32]. This stops attackers from performing a statistical analysis of the data received and inferring personal information from a user's data. For assuring privacy protection, concepts connected to federated learning (FL) are also hot subjects in the research community. FL is a distributed machine learning technique that allows model training for enormous amounts of data to be done locally on the generated source, with each learner in the federation doing the appropriate modeling. Rather than transmitting a raw training dataset, each learner sends his or her local model to an "aggregator" to be combined into a global model. Because FL takes the approach of "bringing the code to the data, rather than the data to the code," it can address critical issues such as data privacy, data ownership, and data localization [33 - 35].

4. RELATED WORKS

Due to the utilization of an open channel for communication. The authors in [36] expressed worry about the security and privacy of Industrial IoT networks. According to the authors, existing approaches may not be suitable in an IIoT-specific setting due to significant overheads. The authors devised a biometric-based privacy-preserving authentication mechanism to counteract unwanted intrusions with minimal overheads. As a two-factor authentication system, the technique employs biometrics and smart cards. To test the protocol's behavior, it was simulated on NS2. After doing formal and informal security analyses, the authors certified their scheme resistant to a variety of assaults.

Despite using two-factor authentication, the technique fails to guard against known key attacks and maintain privacy. The obstacles in establishing security protocols were explored by Li et al. in [37], which included the open nature of the wireless medium and resource-restricted nodes. The authors suggested a three-factor user authentication technique for the WSN-IIoT context that considers these issues. The user's identity, password, and biometrics are the three factors utilized to authenticate. Only if all of the factors provide favorable results will the user be able to view the sensor's data. Although the authors claim their system is immune to impersonation, replay attacks, and other attacks, formal analysis validation is missing.

Because the resource-constrained node transmits and receives a total of 2688 bits during the key exchange procedure, the strategy is wasteful in terms of communication. As a result, the system is unsuitable for resource-constrained IIoT applications. In their paper [38] presented an authentication mechanism for M2M communications in an IIoT environment. According to the authors, traditional techniques cannot be applied in IIoT due to significant overheads that could deplete node resources. As a result, the authors created a novel security model in which only hash and ex-or operations are computed during authentication. The authors declared their approach compute-efficient because of the usage of only a few cryptographic operations. The authors went on to say that their approach has security qualities like session key agreement, and anonymity, and is immune to replay, and man-in-the-meddle (MITM) attacks, among other things. Although the system provides several security benefits, the authors did not do a vulnerability evaluation or formal analysis, therefore the scheme's behavior under compromised settings is uncertain. Furthermore, the technique wastes a lot of energy delivering big mutual authentication and key exchange messages, making it inefficient in terms of energy use. Because of its unpredictable behavior and high energy consumption, the proposed method is unsuitable for IIoT networks.

Xiong et al presented an ECC-based authentication scheme for IIoT in [39]. The authors stressed the importance of an authentication system in WSN to avoid unauthorized access due to the unsecured nature of the medium. Biometrics are used in their scheme to verify the entity's validity. The authors tested their technique on NS3 to see how well it worked. Despite the claimed benefits, it is discovered that the authors did not consider

Denial of Service (DoS) and MITM attacks during the security analysis, which could endanger the network's life. Due to the lack of ciphering and nonce, the system fails to provide privacy and message freshness for all transmitted messages.

Paliwal has stated his concern over data integrity and confidentiality in IIoT networks [40]. The author stressed that sensitive data acquired by sensor nodes in WSN should only be available to those who need it. The article discusses the many available authentication systems as well as their flaws. Hash is used to achieve mutual authentication and key establishment while maintaining identity anonymity. Due to minimal computations and resilience to several significant attacks, the approach is lightweight and efficient. According to the author, the method has undergone formal and informal analysis and is pronounced secure for usage in an IIoT setting. Even though the method is said to be resilient, it does not guarantee privacy. Even though the method does not employ ciphering models, the intensive usage of hashes and the enormous quantity of messages sent overburdens the scheme.

Chang et al. [41] devised an authentication system for WSNs to prevent unauthorized penetrations. Although it is said to be efficient and secure, it is complicated since it runs in two modes. By introducing a smart card-based authentication strategy for WSN, the authors have attempted to address the shortcomings of existing authentication protocols. Their suggested protocol employs two distinct algorithms to achieve two distinct sets of security features. To establish the resilience of their protocol, the authors conducted a formal security analysis using the Real-or-Random (RoR) paradigm. Their first protocol (P1) does not provide complete security solutions, whilst their first protocol (P2) is resource-intensive. Because IoT devices are typically resource-constrained, using this protocol can reduce the devices' and networks' active lifetimes.

In [42], Gope et al. focused on the obstacles to implementing Industrial WSNs (IWSN). The authors designed a new mutual authentication system for IWSN's real-time data access applications, citing security as the most crucial concern. In their approach, the authors used exclusive-or, one-way hash, and physically unclonable functions (PUF), to mention a few. The security of the credentials is the key strength noted in the article, even if the adversary physically captures the sensor nodes. The approach includes important security features, including mutual authentication and integrity. Despite the advantages, the approach requires six messages to complete the session key, which is difficult for devices with limited resources. The number of bits sent in those communications is quite large, which raises the energy consumption threshold even higher. This massive energy usage has the potential to swiftly drain the energy reserves of IIoT nodes. Furthermore, the behavior of the schemes [41, 42] under the effect of a DoS attack is not detected, allowing adversaries to attack IIoT networks via hidden vulnerabilities.

In summary, current approaches are vulnerable to well-known attacks (MITM, Known Key, and DoS, for example). They have large communication and computing costs, making them unsuitable for Industrial IoT networks. The Industrial IoT is a delicate application in which even a little incursion by an unauthorized node can result in significant and irreversible losses. As a result, a secure and efficient key exchange and mutual authentication approach must be used to protect access to the IIoT network. Table 1 summarizes the related works for security issues and applications in IIoT.

There are other research investigations on the security of IIoT based on the 6G network. The authors in [33] offer a high-level overview of the role of trust, security, and privacy in 6G networks and the associated research difficulties. About four essential components of 6G networks, such as real-time intelligent edge computing, distributed artificial intelligence, intelligent radio, and 3D intercoms, this report concisely assesses new study fields and difficulties in security and privacy. Discusses security and privacy concerns with developing technologies such as artificial intelligence (AI), blockchain, quantum communications, Tera Hertz (THz) technology, Visible Light Communication (VLC) technology, and molecular communications. However, the authors in [43] thoroughly examine machine learning and privacy in 6G to accelerate the development of 6G and privacy-protection solutions. At the same time, the authors in [44] discuss unresolved concerns about the applicability of physical layer security (PLS) in 6G systems and provide a complete road map of significant relevant studies on PLS.

Table 1. IIoT Security Related Works Summary

Citations	Security issue	Contributions	shortcomings
Li et al [37]	Authentication for WSN-IIoT	Authentication based on user's identity, password, and biometrics	weak authentication validation
Esfahani et al [38]	M2M security in IIoT	Use hash and ex-or operations during the authentication process	inefficient energy utilization due to big mutual authentication
Xiong et al [39]	Authentication for IIoT Sensor network	avoid unauthorized access due to the unsecured nature of the medium	weak privacy and message freshness
Paliwal [40]	IIoT networks confidentiality	Using Hash for mutual authentication and key establishment	Weak privacy
Chang et al [41]	prevent unauthorized penetrations	formal security analysis using the Real-or-Random (RoR) paradigm	Reduces network device's lifetime
Gope et al [42]	authentication for real-time IIoT	Use exclusive-or, one-way hash, and PUF for mutual authentication	weak against attack via hidden vulnerabilities, and high computation energy

The security and privacy of IIoTs have been discussed by numerous researchers. To create a dependable, accessible, and secure Remote Patient Monitoring (RPM) system in the end, the authors of [45] suggested an integration of the IoT with healthcare facilities that are secure and privacy-preserving. The suggested solution offers end-to-end secure communications, secure RFID-based authentication, and privacy protection. The authors of [46] concentrated on how blockchain can assist 5G network applications in safeguarding execution integrity and proposed a low-cost and simple-to-implement blockchain-based execution protection strategy called NoSneaky. The inventors of [47] suggested a communication protocol that uses only symmetric key-based encryption, which offers incredibly lightweight yet strong encryptions to safeguard data transmissions. To fend off key reset and device capture threats, the symmetric keys created by this protocol are delegated based on a chaotic system, the logistic map.

A blockchain-based deep learning system with two degrees of security and privacy was provided by other authors [48]. To achieve the goal of security and anonymity, a blockchain system is first built in which each participating entity is registered, verified, and then validated utilizing a smart contract-based enhanced Proof of Work. Second, a deep learning system with the Bidirectional Long Short-Term Memory (BiLSTM) for intrusion detection and the Variational Auto Encoder (VAE) technique for privacy is built. The authors in [49] provided a timely discussion of how promissory 6G enabling technologies like artificial intelligence, network softwarization, network slicing, blockchain, edge computing, intelligent reflecting surfaces, backscatter communications, terahertz links, visible light communications, physical layer authentication, and cell-free massive multiple-input multiple-output (MIMO) will play a part in delivering the expected level of security and privacy.

5. PROTOTYPE MODEL (SYSTEM AND ADVERSARY)

The IIoT security based on the system prototype model consists of different IoT components in addition to the adversary model. Fig. 2 shows an internet-connected IIoT network that can be controlled and monitored. The IIoT architecture is made up of IoT sensor nodes installed on machines that connect with the CA and the cloud via a wireless bi-directional link. Through the cloud, the user has access to information. The system prototype model consists of the following devices.

- ***WSN-IIoT network:*** Sensor nodes are installed on machines in the industry. The sensor nodes receive control signals from the operator (e.g., turn on/off the machine), collect data from machines (e.g., production count, machine temperature, pressure, etc.) and wirelessly relay it to the gateway using low-power modules such as Zigbee (IEEE 802.15.4) and Z-Wave (a.k.a, ZW0500).
- ***Gateway***: Typically, a gateway is stationary and powered by the mains. The gateway serves as an intermediary between the smart IoT sensor node, the cloud, and the CA. It supports the IEEE 802.3 and IEEE 802.11 standards for data transmission via the Internet. The gateway authenticates the IIoT network's nodes before transferring their data to the cloud and vice versa.
- ***Certification authority:*** The certification authority (e.g., Symantec, GeoTrust, and others) builds a database of the network's nodes and uses it to undertake mutual authentication before giving certificates to nodes. Each sensor node receives a unique implicit certificate from the CA, which they must use to create public and private keys.

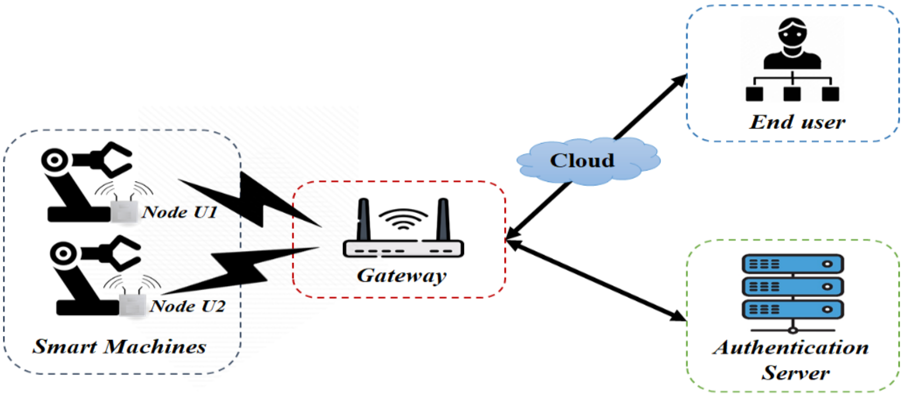


Fig. 2. An IIoT system model based on a mutual authentication key exchange method

The Dolev-Yao adversary model recommended in [41, 50] has been used in the suggested approach. According to the threat model, the adversary can uncover the industrial network's flaws, which can then be exploited to exploit the industries' potential resources. Consider an IIoT-enabled smart automobile manufacturing business [51], where sensor nodes are used to monitor and control robotic arm activities, manage logistics, and identify raw material requirements at the warehouse, among other things. According to the Dolev-Yao adversary model, robotic industrial machines (nodes), logistics and warehouse network devices (gateway), and other IIoT devices are under threat. In the IIoT, an adversary can listen to all conversations between industrial nodes, gateways, and CA.

An adversary can collect, modify, and replay network signals to gain privileged access to industrial robotic arms (e.g., welding, painting, transportation, and assembling), among other things. In addition, an adversary can pose as a legal industrial node to steal data from RFID tags. Physical capture of smart industry devices (nodes and gateways) is not conceivable because they are secured with physical locks and monitored by surveillance cameras. The adversary may attempt to change the lifetime of the expired authenticator to gain unauthorized access to the industrial network and introduce malware into the industry's computerized production units. Furthermore, the attacker can intercept data sent between network entities to obtain security parameters that can be used to generate future secret keys, activate driverless cars, and so on.

The adversary can create and inject new messages through the network to perform a DoS attack that prevents control orders from being sent to industrial machinery (e.g., warehouse storage sequencing error). To summarize, the opponent can obstruct the smooth and secure operation of production units, warehouses, and logistics, among other things. Financial and reputational harm, company interruption, and lower efficiency are all possible outcomes of hostile attacks.

6. PROPOSED METHODOLOGY

Multi-variable identifications (M-VIDs), which cannot be easily associated with the ID or tracked, must be assigned to defend against attacks related to identification. To satisfy these needs, this technique switches the fixed ID identification out for the regularly changing M-VID identifiers. Before sending the range to the UE, the serving network (SN) assigns a range of M-VID IDs to the User (D). The SN then starts the ID relocation process and gives the user two M-VID values, S and L, which stand for the range's start and length M-VID values. S stands for the range's start point, and L for the range's length. It is up to the network operator to specify the length K. The user reads the allocated range D as follows: The largest M-VID in D is (S+L), whereas the smallest M-VID in D is S. The SN then randomly produces a new M-VID value between S and S+L whenever it needs to identify the user and adds it in the identification message that will be sent to the user. The user equipment also knows that the M-VID utilized for identification should remain between S+L and S.

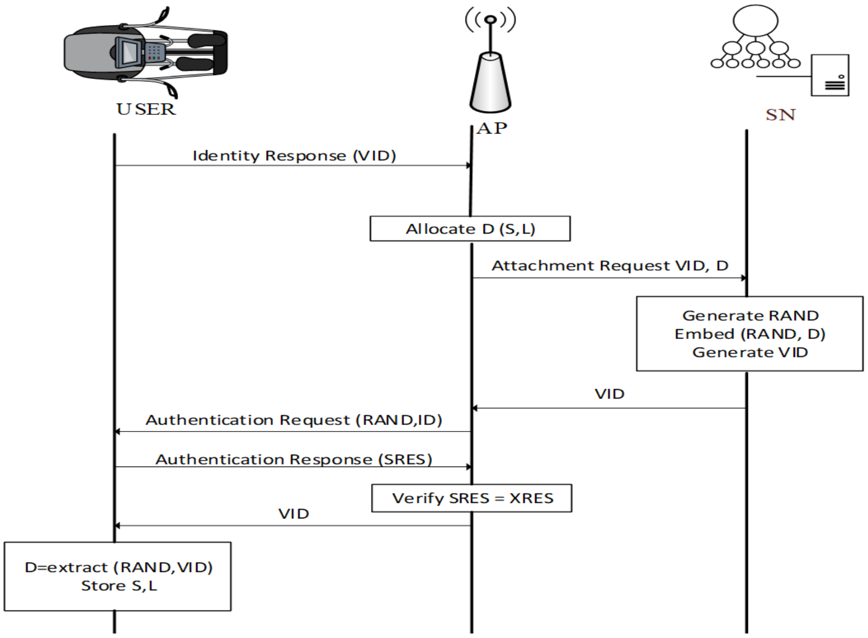


Fig. 3. The essential steps of sending and assigning of ID range to the user.

The SN incorporates the newly created M-VID value among S+L and S into the identification message that will be sent to the user. When the SN wants to identify the user, this occurs. The user checks the incoming M-VID to see if it falls between S+L and L or not. If the received M-VID falls within the proper range, the user may respond and begin the service request procedure; if not, the user discards the message requesting identification. Fig. 3 depicts the allocation procedure and the user receiving the M-VIDs range.

To implement the proposed solution some modifications must be executed in SN and user. As illustrated in the following subsections, we provide our proposed algorithms for both SN and user end.

6.1 The Proposed Algorithm in SN

The proposed solution suggested values for all identifiers as shown in Table 2. A table called P-table contains M-VIDs for all users within its service region that have been added to the SN storage. One user's M-VIDs are stored in a tuple of P-tables, which include the fields VID, S, T, V, and V. The start and length M-VID values in the user-assigned range are denoted by S and L, respectively. The T represents the M-VID value that was most recently used to identify the user, whereas the V represents the M-VID value that the user most recently used to submit a service request. The SN furthermore maintains a list of M-VID ranges known as VID-pool. The VID-pool is a table with columns S, L, and STATUS, as illustrated in Table 2. M-VID ranges' start and length M-TMSI values are stored in variables S and L, respectively. The STATUS next to each range indicates whether or not the range is available for use. When STATUS is set to 0, it means that the relevant range is available for use. The associated STATUS will be 1 for the given range.

Table 2**.** M-VID values in the proposed solution

a- VID Pool

S	*L*	*STATUS*
S_1	L_1	*1*
...	...	…
Si	L_i	1
...	...	…
S_K	L_K	0
...	...	…
Sn	*Ln*	0

b- P-table

VID	*S*	*L*	*T*	*V*
VID_1	S_1	L_1	T_1	V_1
…	...	...	…	…
…	...	...	…	…
VIDi	*Si*	*Li*	*Ti*	*Vi*
…	...	...	…	…
…	...	...	…	…
VID_K	S_K	L_K	T_K	V_K

The proposed scheme can be described using two phases: Setup and Manage M-TMSIs.

A. Setup Phase (The Initial Allocation)

The initial M-VIDs range allocation to users within the SN's service region is carried out during the setup phase. Only the initial execution of the Setup phase is performed, and it must be successful before the management phase is launched. The following are the main steps in the Setup phase:

- VID-pool information initialization using M-VIDs: The SN executes the Initialize-Pool algorithm to initialize the M-VID-pool with the M-VID range bounds.
- Give the users access to the M-VID ranges: Within its service area, the SN executes the Allocate Range algorithm for each user.
- Provide the M-VID ranges to the users: The SN provides the bounds of the M-VID ranges assigned to the concerned users.

B. Manage Phase (Monitor and Control)

As depicted in Algorithm 1, the manage phase entails continuing actions and processes that include monitoring the M-VID-related service requests made by the user, the SN, and other SNs, and appropriately modifying the M-VID data at the SN. Through several methods for M-TMSI range allocation, re-allocation, and de-allocation during the Manage phase, the SN manages the M-VID identities and preserves the consistency of the contents of the P-table and the M-VID-pool.

• M-VID range Allocation: The SN allocates a new M-VID range D to the user after successful authentication runs.

• M-VID range Re-Allocation: After a successful run of the Tracking Area Update (TAU) procedure, the SN determines whether to replace the M-VID range that is currently allocated to the user or to keep it. If the range currently allocated to the user will cause an M-VID collision, the SN replaces it with a new range.

• M-VID range De-Allocation: The SN de-allocates the M-VID range allocated to a user after a successful request.

• M-VID Validation: When a user sends a request including an M-VID identifier to the SN, the latter verifies that the request is initiated by a genuine user using the Validate Request algorithm.

6.2 The Proposed Algorithm for User

The suggested scheme needs the user to be expanded to store the following four values: S_{User}, L_{User}, T_{User}, and V_{User}. The bounds of the M-VID range provided by the SN are stored in the S_{User} and L_{User}. The M-VID identities that the user most recently sent and received are kept in the T_{User} and the V_{User} respectively. Modifications about VID relocation, identification, and service request processes should also be made to the user's functionality.

The user confirms that the embedded M-VID value within the identification message is within the appropriate range (VID is between S_{User} and S_{User} +L_{User}) and is distinct from the M-VID that was last delivered or received by the user after receiving an identification message request from the SN (T_{User} or V_{User}). If so, the user replies by submitting a service request and updating its T_{User} to the newly arriving VID identity. If not, the message request is ignored. Algorithm 2 shows the M-VID validation process.

Algorithm 1: The Manage phase algorithm

```
Input: The VID or ID identifiers of the user involved in the request and service request code
while true do
   if request ='An Attachment' then
      call Allocate-Range (VID)
      call VID-Relocation-Procedure
end if
   if request ='Tracking Area Update TAU' then
      call ReAllocate-Range (VID)
      call ID-Relocation-Procedure
   end if
   if request ='Request from the SN to forget about the User" then
      call DeAllocate-Range (VID)
   end if
   if request ='Identification the User' then
      call Identification-User (VID)
   end if
   if timer ='0' then
   for each User whose timer is expired do
         call ReAllocate-Range (VID)
         call ID-Relocation-Procedure
      end for
   end if
   if request ='Radio Resource Channel request with ID identifier' then
         call Validate-Request (ID)
   if Auth = true then
            process the request
```

```
        else
                discard the request
            end if
        end if
    end while
```

Algorithm 2: Identifying message validation algorithm

```
Input: identifying message including the ID (VID) received from serving network
if ($S_{User} \leq VID \leq S_{User} + L_{User}$)
    if ($VID \neq T_{User}$ & $VID \neq V_{User}$)
        update $V_{User} = VID$
        initiate a service request
    else
        discard the request
    end if
else
    discard the request
end if
```

If a user certifies that it was the intended recipient of the identification message sent by the SN, the user starts a service request. To start a service request, the user first creates a new M-VID value at random (VIDUser), inserts it into the message to the SN, and changes VIDUser to TUser. Algorithm 3 describes the stages involved in a service request.

Algorithm 3: Service Request Algorithm

```
create a random fresh $M_U$ such that:
$S_U \leq M_U \leq (S_U + L_U)$,
$M_U \neq T_U$, and
$M_U \neq V_U$
update $T_U = M_U$
initiate service request
```

7. ANALYSIS AND DISCUSSION

In the current IIoT architecture, a user is given an ID identifier to be able to be identified specifically throughout the identification process. The user's ID is always included in the identifying request message and sent to the user whenever the providing network wants to identify an idle user. The issue is that the allocated ID is kept for a long enough time for an attacker to connect it to the permanent identification ID of the user and use it to attach identifying communications to that user.

As a result, the current identification process is not secure against user link-ability attacks. The properties of ID identifiers and the ID allocation mechanism are recommended to be improved, adding security performance and protecting against link-ability attacks. Each time a user is identified, a random VID identification is used, which ensures that an observer cannot connect the identifying request to the same user.

7.1 The Key Features

In our proposed approach, the user is only required to perform a minimal amount of computation, with the serving network bearing the rest of the burden. We assert that the overhead is little since the SN has limitless computing capacity. We additionally assert that the user's calculation overhead is minimal. The delay time in SN and the user comparing by utilizing ID rose slightly due to the VID changing in every identification and random access. It nonetheless outperformed encryption techniques like ENC-ID. The VID enhanced the characteristics of ID identifiers by replacing it with the VIDs which added a high level of user privacy in IIoT networks. A new random VID is generated and consumed whenever a user is requested to be identified to the SN. This guarantees that an observer cannot link the VIDs to a certain user, and hence prevents against tracking the user. As the VID changes in every request, the delay time increases slightly in the network and the user compared by using ID. However, it was better than an encryption method like ENC-ID. The delay time on SN and user by using the ID, VID, and ENC-ID.

Figures 4 and 5 display the delay time on SN and user utilizing the ID, VID, and ENC-ID. The identification process takes a little longer when M-VIDs are used than the normal technique because the identification (ID) is sent in clear text. While the identifying process takes longer with the encryption method. Because encryption methods require algorithms to encrypt and decrypt the ID with every identification operation, there was a greater time delay or overhead. Encryption and decryption tasks will take longer to complete in user and SN. In addition to that, each identifying procedure requires more time for the generation of encryption keys.

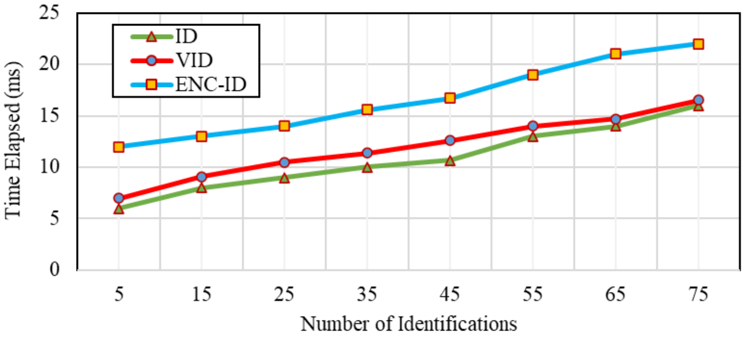


Fig. 4: User identification overhead on SN.

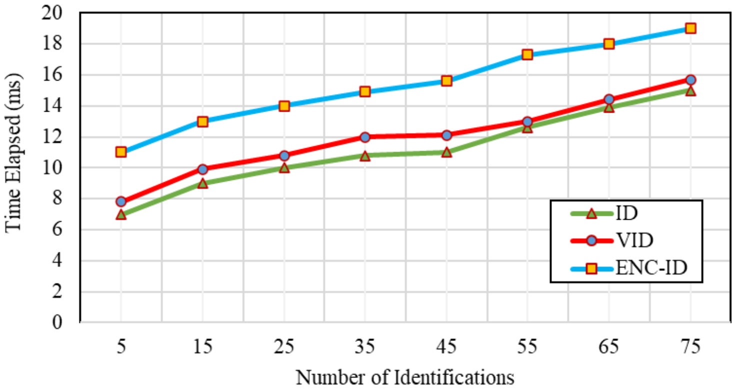


Fig. 5: User identification overhead on User.

Considering the features of system impact and Compatibility with IoT architecture, the proposed solution gives a minimal system impact, which is transparent to the intermediary networks because it does not call for modifications to the messages or the messaging infrastructure. Due to the minimum changes, it requires of the network parties, the solution can easily be compatible with the current IIoT architecture.

7.2 Security Analysis

The security of the solution is examined in this part in terms of unlinkability, anonymity, and untraceability.

A. *User Unlink-ability*

Linkability is the potential for connecting different user identities. By making IIoT networks unlinkable, the proposed technique eliminates user linkability and defends users against tracking attacks. Instead of being given a permanent ID that can be traced and associated with a specific user, the user is given a series of temporary identities, or VIDs. As seen in Fig. 6, a new random VID is generated and used each time a user requests to be recognized by the network. This ensures that a viewer cannot connect the VIDs to a specific user, preventing the viewer from tracking the user.

B. User Anonymity

The suggested system offers a high level of confidence in preserving user identification. Since the ID is only accessible by the NS and the user and no other party on the network is aware of it, an attacker cannot know it. The ID is also never utilized or communicated. Because the NS changes the VID before being sent to the user, there is no way for an attacker to determine the VID given to a specific user. Until a user uses their VID for identification, the attacker is not made aware of their VID. It is important to note that the attacker cannot benefit from knowing a specific VID. The approach used by the suggested scheme about VID selection grants a user the right to protect their user anonymity and hinders attackers from doing so. The user can only use a VID once, therefore as soon as the network successfully identifies them, they are given a brand-new VID that is distinct from the one they were previously using. The brand-new VID given to the user is chosen randomly and has nothing to do with the VID that they utilized most recently. VIDs allocated to a specific user appear to an attacker to be random bit streams that cannot be connected to a specific user. As a result, the attacker is unable to identify the target user, and Fig. 6 illustrates the provision of the highest level of identity anonymity.

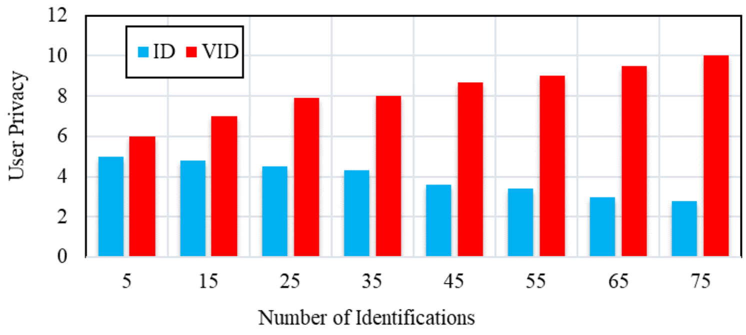


Fig. 6: User privacy comparison between ID and VID.

C. User Un-traceability

Traceability is the ability to track previous identity requests and responses coming from the same subscriber. The proposed method improves the properties of the pseudonyms and the methods for allocating them, which prevents user traceability and defends users from tracking attacks (TIDs). This makes it challenging for an observer to distinguish between identification requests and responses sent to the same user because the pseudonyms exchanged in the network appear random and unrelated from the observer's point of view. As a result, the user's untraceability is provided and the observer is unable to recognize the user's previous identification requests and responses.

8. CONCLUSION

The issue of safeguarding the privacy of the identifying procedure in the IIoT network is addressed in this study with a practical solution. Through a secure identification technique that enables a user to be uniquely identified by the network while remaining anonymous within the network, the identifying procedure privacy is maintained, preventing adversaries from being able to monitor and identify the user. The benefit of the solution is that it is simple to integrate into the existing architecture and is compatible with current IIoT technology standards. With minimum changes at both the network and the user level, low computing overhead on the part of the network, and negligible calculation overhead on the part of the user, the proposed method protects the identifying procedure privacy in IIoT and ensures user un-traceability and unlink-ability.

REFERENCES

[1]. Peter, O., Pradhan, A., & Mbohwa, C. Industrial internet of things (IIoT): opportunities, challenges, and requirements in manufacturing businesses in emerging economies. Procedia Computer Science, (2023), 217, 856-865, https://doi.org/10.1016/j.procs.2022.12.282

[2]. Kumar, R., Rani, S., & Awadh, M. A. Exploring the application sphere of the internet of things in industry 4.0: a review, bibliometric and content analysis. Sensors, (2022), 22(11), 4276, https://doi.org/10.3390/s22114276.

[3]. Garrido, G. M., Sedlmeir, J., Uludağ, Ö., Alaoui, I. S., Luckow, A., & Matthes, F. Revealing the landscape of privacy-enhancing technologies in the context of data markets for the IoT: A systematic literature review. Journal of Network and Computer Applications, (2022), 207, 103465, https://doi.org/10.1016/j.jnca.2022.103465

[4]. Ali, A., Al-Rimy, B. A. S., Alsubaei, F. S., Almazroi, A. A., & Almazroi, A. A. HealthLock: Blockchain-Based Privacy Preservation Using Homomorphic Encryption in Internet of Things Healthcare Applications. Sensors, (2023), 23(15), 6762, https://doi.org/10.3390/s23156762

[5]. Rizi, M. H. P., & Seno, S. A. H. A systematic review of technologies and solutions to improve security and privacy protection of citizens in the smart city. Internet of Things, (2022), 20, 100584, https://doi.org/10.1016/j.iot.2022.100584

[6]. Kamdjou, H. M., Baudry, D., Havard, V., & Ouchani, S. Resource-Constrained eXtended Reality Operated with Digital Twin in Industrial Internet of Things. IEEE Open Journal of the Communications Society, (2024), https://doi.org/10.1109/OJCOMS.2024.3356508

[7]. Kamarudin, N. H., Suhaimi, N. H. S., Nor Rashid, F. A., Khalid, M. N. A., & Mohd Ali, F. Exploring Authentication Paradigms in the Internet of Things: A Comprehensive Scoping Review. Symmetry, (2024), 16(2), 171, https://doi.org/10.3390/sym16020171

[8]. Mengistu, T. M., Kim, T., & Lin, J. W. A Survey on Heterogeneity Taxonomy, Security and Privacy Preservation in the Integration of IoT, Wireless Sensor Networks and Federated Learning. Sensors, (2024), 24(3), 968, https://doi.org/10.3390/s24030968

[9.] Alotaibi, B. A survey on industrial Internet of Things security: Requirements, attacks, AI-based solutions, and edge computing opportunities. Sensors, (2023), 23(17), 7470, https://doi.org/10.3390/s23177470

[10]. Mohsan, S. A. H., & Li, Y. A Contemporary Survey on 6G Wireless Networks: Potentials, Recent Advances, Technical Challenges and Future Trends. arXiv preprint arXiv:2306.08265, (2023), https://doi.org/10.48550/arXiv.2306.08265

[11]. Yazici, İ., Shayea, I., & Din, J. A survey of applications of artificial intelligence and machine learning in future mobile networks-enabled systems. Engineering Science and Technology, an International Journal, (2023), 44, 101455, https://doi.org/10.1016/j.jestch.2023.101455

[12]. Huda Mahmood, Nurul, et al. "Six Key Enablers for Machine Type Communication in 6G." arXiv e-prints (2019): arXiv-1903, https://doi.org/10.48550/arXiv.1903.05406

[13]. Hasan, M. K., et al. "Inter-cell interference coordination in LTE-A HetNets: A survey on self organizing approaches." 2013 International Conference on Computing, Electrical and Electronic Engineering (ICCEEE). IEEE, 2013, https://doi.org/10.1109/ICCEEE.2013.6633932

[14]. Saeed, M. M. A., Saeed, R. A., & Ahmed, Z. E. (2024). Data Security and Privacy in the Age of AI and Digital Twins. In Digital Twin Technology and AI Implementations in Future-Focused Businesses (pp. 99-124). IGI Global, https://doi.org/10.4018/979-8-3693-1818-8.ch008

[15]. Saeed, Mamoon M., et al. "A novel variable pseudonym scheme for preserving privacy user location in 5G networks." Security and Communication Networks 2022 (2022), https://doi.org/10.1155/2022/7487600

[16]. Hasan, Mohammad Kamrul, et al. "Evolution of industry and blockchain era: monitoring price hike and corruption using BIoT for smart government and industry 4.0." IEEE Transactions on Industrial Informatics 18.12 (2022): 9153-9161, https://doi.org/10.1109/TII.2022.3164066

[17]. Strinati, Emilio Calvanese, et al. "6G: The next frontier: From holographic messaging to artificial intelligence using subterahertz and visible light communication." IEEE Vehicular Technology Magazine 14.3 (2019): 42-50, https://doi.org/10.1109/MVT.2019.2921162

[18]. Tariq, Faisal, et al. "A speculative study on 6G." IEEE Wireless Communications 27.4 (2020): 118-125, DOI: 10.1109/MWC.001.1900488

[19]. Van Der Zwaag, Klaas Minne, et al. "A manchester-ook visible light communication system for patient monitoring in intensive care units." IEEE Access 9 (2021): 104217-104226, https://doi.org/10.1109/ACCESS.2021.3099462

[20]. Saeed, Mamoon M., et al. "A comprehensive review on the users' identity privacy for 5G networks." IET Communications 16.5 (2022): 384-399, https://doi.org/10.1049/cmu2.12327

[21]. Saeed, Mamoon M., et al. "Task Reverse Offloading with Deep Reinforcement Learning in Multi-Access Edge Computing." 2023 9th International Conference on Computer and Communication Engineering (ICCCE). IEEE, 2023, https://doi.org/10.1109/ICCCE58854.2023.10246081

[22]. Ahmed, Zeinab E., et al. "Mobility Management Enhancement in Smart Cities using Software Defined Networks." Scientific African (2023): e01932, https://doi.org/10.1016/j.sciaf.2023.e01932

[23]. Amanlou, Sanaz, Mohammad Kamrul Hasan, and Khairul Azmi Abu Bakar. "Lightweight and secure authentication scheme for IoT network based on publish–subscribe fog computing model." Computer Networks 199 (2021): 108465, https://doi.org/10.1016/j.comnet.2021.108465

[24] Huda Mahmood, Nurul, et al. "Six Key Enablers for Machine Type Communication in 6G." arXiv e-prints (2019): arXiv-1903, https://doi.org/10.48550/arXiv.1903.05406

[25]. Saeed, Mamoon M., et al. "Attacks Detection in 6G Wireless Networks using Machine Learning." 2023 9th International Conference on Computer and Communication Engineering (ICCCE). IEEE, 2023, https://doi.org/10.1109/ICCCE58854.2023.10246078

[26]. Saeed, Mamoon M., et al. "Green Machine Learning Approach for QoS Improvement in Cellular Communications." 2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA). IEEE, 2022, https://doi.org/10.1109/MI-STA54861.2022.9837585

[27]. Saeed, Mamoon M., et al. "Anomaly Detection in 6G Networks Using Machine Learning Methods." Electronics 12.15 (2023): 3300, https://doi.org/10.3390/electronics12153300

[28]. Muthana, Abdulrahman A., and Mamoon M. Saeed. "Analysis of user identity privacy in LTE and proposed solution." International Journal of Computer Network and Information Security 9.1 (2017): 54, https://doi.org/.5815/ijcnis.2017.01.07

[29]. Saeed, Mamoon M., Rashid A. Saeed, and Elsadig Saeid. "Preserving privacy of paging procedure in 5th G using identity-division multiplexing." 2019 First International Conference of Intelligent Computing and Engineering (ICOICE). IEEE, 2019, https://doi.org/10.1109/ICOICE48418.2019.9035167

[30]. Saeed, Mamoon M., et al. "Preserving Privacy of User Identity Based on Pseudonym Variable in 5G." Computers, Materials & Continua 70.3 (2022), https://doi.org/10.32604/cmc.2022.017338

[31]. Wang, Qixu, et al. "PCP: A privacy-preserving content-based publish–subscribe scheme with differential privacy in fog computing." IEEE Access 5 (2017): 17962-17974, https://doi.org/10.1109/ACCESS.2017.2748956

[32]. Bonawitz, Keith, et al. "Towards federated learning at scale: System design." Proceedings of machine learning and systems 1 (2019): 374-388, https://doi.org/10.48550/arXiv.1902.01046

[33]. Niknam, Solmaz, Harpreet S. Dhillon, and Jeffrey H. Reed. "Federated learning for wireless communications: Motivation, opportunities, and challenges." IEEE Communications Magazine 58.6 (2020): 46-51, https://doi.org/10.1109/MCOM.001.1900461

[34]. Ylianttila, Mika, et al. "6G white paper: Research challenges for trust, security and privacy." arXiv preprint arXiv:2004.11665 (2020), https://doi.org/10.48550/arXiv.2004.116

[35]. Das, Ashok Kumar, et al. "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment." IEEE Internet of Things Journal 5.6 (2018): 4900-4913, https://doi.org/10.1109/JIOT.2018.2877690

[36]. Saeed, R. A., Saeed, M. M., Ahmed, Z. E., & Hashim, A. H. (2024). Enhancing Medical Services Through Machine Learning and UAV Technology: Applications and Benefits. In Applications of Machine Learning in UAV Networks (pp. 307-343). IGI Global https://doi.org/10.4018/979-8-3693-0578-2.ch012

[37]. Li, Xiong, et al. "A robust and energy efficient authentication protocol for industrial internet of things." IEEE Internet of Things Journal 5.3 (2017): 1606-1615, https://doi.org/10.1109/JIOT.2017.2787800

[38]. Esfahani, Alireza, et al. "A lightweight authentication mechanism for M2M communications in industrial IoT environment." IEEE Internet of Things Journal 6.1 (2017): 288-296, https://doi.org/10.1109/JIOT.2017.2737630

[39]. Xiong Li, et al. "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things." IEEE Transactions on Industrial Informatics 14.8 (2017): 3599-3609, https://doi.org/10.1109/TII.2017.2773666

[40]. Paliwal, Swapnil. "Hash-based conditional privacy preserving authentication and key exchange protocol suitable for industrial internet of things." IEEE Access 7 (2019): 136073-136093, https://doi.org/10.1109/ACCESS.2019.2941701

[41]. Chang, Chin-Chen, and Hai-Duong Le. "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks." IEEE Transactions on wireless communications 15.1 (2015): 357-366, https://doi.org/10.1109/TWC.2015.2473165

[42]. Gope, Prosanta, et al. "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks." IEEE transactions on industrial informatics 15.9 (2019): 4957-4968, https://doi.org/10.1109/TII.2019.2895030

[43]. Sun, Yuanyuan, et al. "When machine learning meets privacy in 6G: A survey." IEEE Communications Surveys & Tutorials 22.4 (2020): 2694-2724, https://doi.org/10.1109/COMST.2020.3011561

[44]. Shakiba-Herfeh, Mahdi, Arsenia Chorti, and H. Vincent Poor. "Physical layer security: Authentication, integrity, and confidentiality." Physical layer security (2021): 129-150, https://doi.org/10.1007/978-3-030-55366-1_6

[45]. Ahmed, Mohammed Imtyaz, and Govindaraj Kannan. "Secure and lightweight privacy preserving Internet of things integration for remote patient monitoring." Journal of King Saud University-Computer and Information Sciences 34.9 (2022): 6895-6908, https://doi.org/10.1016/j.jksuci.2021.07.016

[46]. Chiu, Wei-Yang, Weizhi Meng, and Chunpeng Ge. "NoSneaky: A Blockchain-Based Execution Integrity Protection Scheme in Industry 4.0." IEEE Transactions on Industrial Informatics (2022), https://doi.org/10.1109/TII.2022.3215606

[47]. Luo, Xi, et al. "A lightweight privacy-preserving communication protocol for heterogeneous IoT environment." IEEE Access 8 (2020): 67192-67204, https://doi.org/10.1109/ACCESS.2020.2978525

[48]. Almaiah, Mohammed Amin, et al. "A lightweight hybrid deep learning privacy preserving model for FC-based industrial internet of medical things." Sensors 22.6 (2022): 2112, https://doi.org/10.3390/s22062112

[49]. Osorio, Diana Pamela Moya, et al. "Towards 6G-enabled internet of vehicles: Security and privacy." IEEE Open Journal of the Communications Society 3 (2022): 82-105, https://doi.org/10.1109/OJCOMS.2022.3143098

[50]. Wang, Ding, Wenting Li, and Ping Wang. "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks." IEEE Transactions on Industrial Informatics 14.9 (2018): 4081-4092, https://doi.org/10.1109/TII.2018.2834351

[51]. Sooriakumaran, Prasanna, et al. "A multinational, multi-institutional study comparing positive surgical margin rates among 22 393 open, laparoscopic, and robot-assisted radical prostatectomy patients." European urology 66.3 (2014): 450-456, https://doi.org/10.1016/j.eururo.2013.11.018

ASIAN JOURNAL OF ELECTRICAL AND ELECTRONIC ENGINEERING
Vol. 4 No. 2 2024
E-ISSN: 2785-8189

Evaluation of Classification Algorithms in Tracing Malicious Telephone Numbers

Van Vuong Ngo*

Hanoi, Vietnam

*Corresponding author: vanvuong.ngo.vt@gmail.com

(Received:7 August 2024; Accepted: 27 August 2024)

Abstract— Mobile phones and telecommunications networks have recently played an important role in modern society. They are dispensable parts of our lives as they facilitate the way we communicate. However, apart from their benefit, their proliferation has some drawbacks as telephone networks can be exploited. For example, commercial calls can be made repeatedly to advertise companies' products. These calls annoy customers because they promote products without considering customers' interests. These unexpected calls not only cause a negative impact on the networks but also disturb mobile phone users. To confront this problem, the network administrators need some methods to detect the phone numbers that are used to make the harassment. Therefore, we proposed a solution based on machine learning classification models. Then the performance of some models, namely K-Nearest Neighbors, Decision Tree, and Logistic Regression, is compared. By applying the machine learning models, network administrators can identify and restrict malicious telephone numbers.

Keywords: *Telecommunications, Classification, Machine learning, Network.*

1. INTRODUCTION

This section introduces telecommunications networks and the issue of telephone harassment. It also shows some information about machine learning and its algorithms.

1.1 Telephone Harassment

In recent years, mobile phones have become ubiquitous. With the advancement of technology, mobile phones help us in many aspects of our lives. For example, they enhance the ability to communicate frequently. Besides, telecommunications systems are expanding rapidly. Apart from 4G networks [1], the telecommunications vendors are also researching on 5G or 6G networks [2][3]. Fig. 1 depicts a simple telecommunications network with many subsystems such as Evolved Packet Core (EPC) [4], Public Switching Telephony Network (PSTN) [5], or 5G New Radio (5G NR) [6].

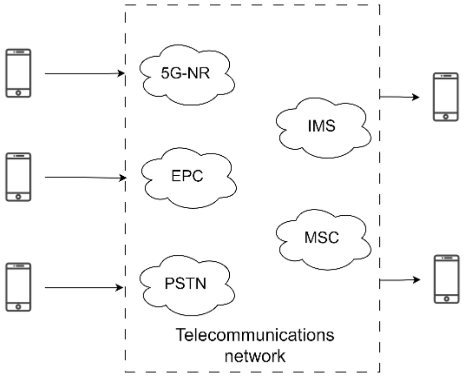


Fig. 1. A telecommunications system.

DOI: https://doi.org/10.69955/ajoeee.2024.v4i2.64

Nevertheless, the expansion of telecommunications systems and personal devices brings some things that could be improved. One of the issues is telephone harassment. Telephone harassment can come in many forms. They can be advertising calls to introduce some products which the customers do not care about. Another form is disruptive calls that clog the hotlines of companies. This type of attack is often performed automatically by pre-programmed software. These malicious calls affect telecommunications networks and other mobile phone users.

1.2 Machine learning algorithms

There are some categories of machine learning algorithms: supervised algorithms, unsupervised algorithms... [7]. Supervised algorithms are divided into two types: classification and regression. While regression predicts the output values based on the input data, classification categorizes outputs into predefined values or classes. For example, predicting the price of a house is an example of regression, while determining whether an animal in a photo is a cat or a dog is an example of classification. Some classification algorithms are shown in Fig. 2.

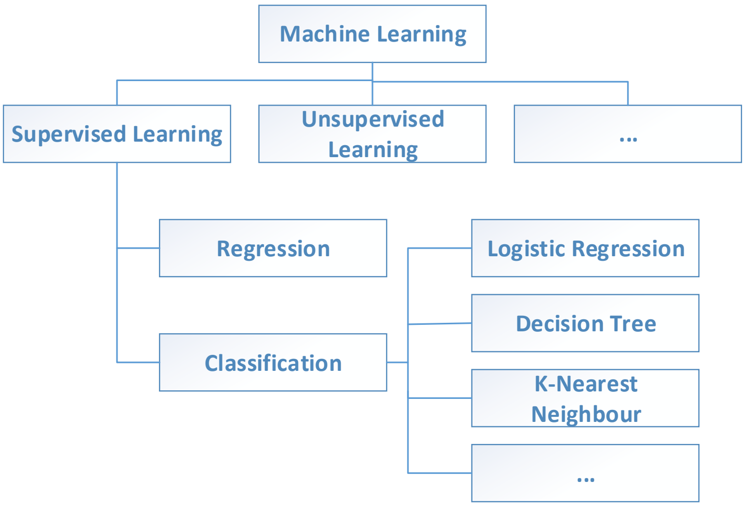


Fig. 2. Machine learning categories.

2. MOTIVATION AND RELATED WORKS

The advent of machine learning (ML) applications in telecommunications has recently attracted researchers' interest. In [8], the authors proposed a multi-layer model to determine the reasons for call failure. By analyzing the Call Detail Records (CDRs) [9], the model can classify problems into different failure categories, such as Charging Failure or Media Failure.

Another novel topic is Security using Machine Learning. An intrusion detection method has been developed for multimedia platforms [10]. This intrusion detection method protects against flooding attacks, which can cause network congestion. Sammer [11] also proposed an ML-based approach for intrusion detection in Mobile Ad hoc Networks (MANETs).

Regarding other topics, a manuscript [12] discusses the application of ML in analyzing customer behavior based on features such as age, gender, or annual income. Other ML applications were introduced in [13][14], which are about detecting fake data or predicting cryptocurrency prices.

Therefore, we came up with the idea that Machine Learning models can mitigate the telephone harassment problem. By Applying machine learning models, network administrators can identify the malicious telephone numbers that cause problems for networks and users.

3. CLASSIFICATION MODELS FOR TRACING MALICIOUS TELEPHONE NUMBERS

3.1 Logistic Regression

This classification model is based on the neural network concept [15]. The neural network consists of three layers: the input layer, the hidden layer, and the output layer. Each layer has some nodes which have their own weights. When the inputs are forwarded to a layer, the inputs are multiplied by adjusted weights to produce the outputs of that layer. Finally, the output layer applies a binary activation function that can predict whether the result is 0 or 1. In the case of our study, the activation function suggests whether a mobile phone number makes telephone harassment or not. The Sigmoid function [16] is a suitable choice for the activation function because it is monotonic and its values range between 0 and 1. The Sigmoid function is defined as formula (1) and depicted in Fig. 3.

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

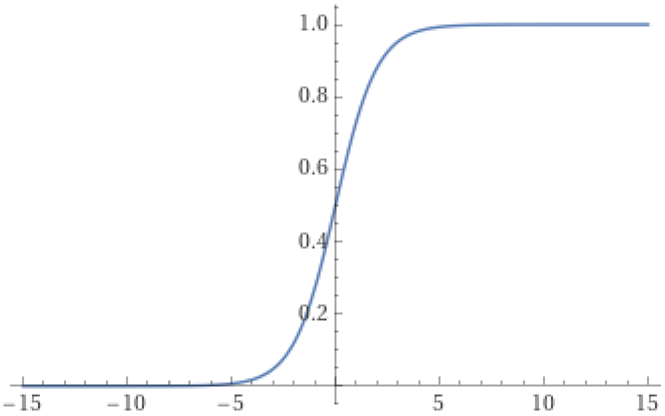


Fig. 3. The Sigmoid function

Fig. 4 depicts a neural network for the classification model. The input of the neural network is the matrix [N*M]. Each row of the input matrix represents a phone number and its attributes. While N is the number of mobile phone numbers that need to be examined, M is the number of attributes. For instance, if the number of calls per day and the average call duration are chosen as attributes, then M equals 2 in this case. It can be predicted that mobile phone numbers that make a significant number of calls with short call duration are the cause of telephone harassment. At the final layer, the output is the matrix [N*1]. Each row of the output matrix takes the binary value 0 or 1, which indicates whether the mobile phone number of this row makes telephone harassment or not. Thanks to this result, network administrators can impose restrictions on phone numbers that cause harassment. Fig. 5 shows an example of the evaluation with the neural network.

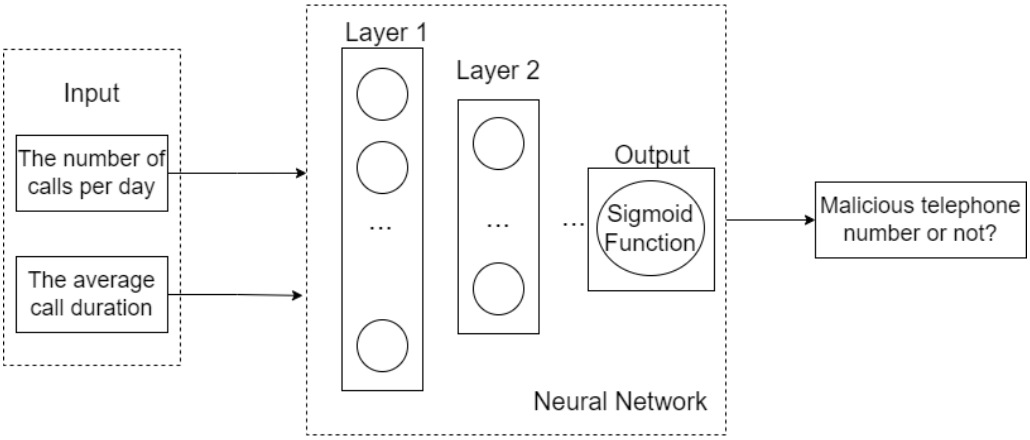
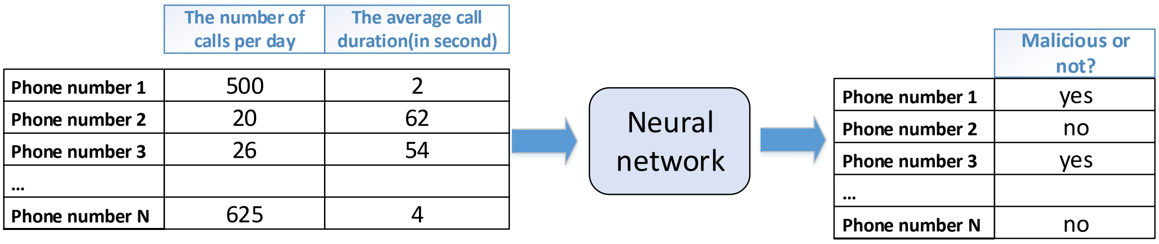

Input
The number of calls per day
The average call duration
Layer 1
Layer 2
...
Output
Sigmoid Function
Neural Network
Malicious telephone number or not?

Fig. 4. The neural network model



	The number of calls per day	The average call duration(in second)
Phone number 1	500	2
Phone number 2	20	62
Phone number 3	26	54
…		
Phone number N	625	4

	Malicious or not?
Phone number 1	yes
Phone number 2	no
Phone number 3	yes
…	
Phone number N	no

Fig. 5. An example of the evaluation with the neural network

3.2 *K-nearest Neighbors*

K-nearest neighbors (KNN) is a popular supervised machine learning algorithm [17]. The idea of KNN is that the characteristic of a data point is similar to its closest neighboring points. K is the number of nearest neighbors to use. The distance between the given point and other points is calculated to determine which points are closest to a given data point. The distance between these points is calculated using Euclidean distance formula as follows:

$$d(x,y) = \sqrt{(y-x)^2} \qquad (2)$$

Fig. 6 illustrates the KNN algorithm. Fig. 6a shows the case where K = 1, and the data point is predicted as class 1 because its closest point belongs to class 1. In Fig. 6b, for K =3, among the three closest points of the given point, there are two class 2 points and one class 1 point. Therefore, the given point is predicted to be class 2. Applying the KNN algorithm to the problem of malicious telephone numbers, the vertical axis can represent the number of calls per day, whereas the horizontal axis can represent the average call duration, as can be seen in Fig. 7.

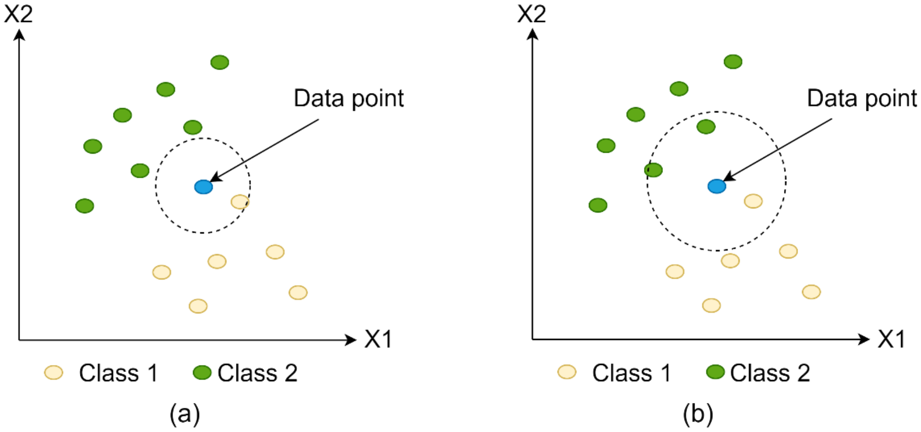


Fig. 6. The KNN algorithm

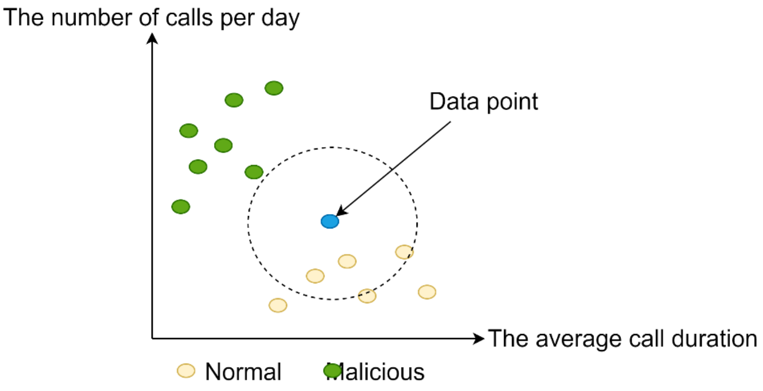


Fig. 7. The KNN diagram

3.3 *Decision Tree*

Decision Tree is a tree-structured classifier that is preferred for solving classification problems [18]. The initial node is the root of the tree, branches represent the decision rules, and leaves are the outcomes. Fig. 8 shows a model of a decision tree.

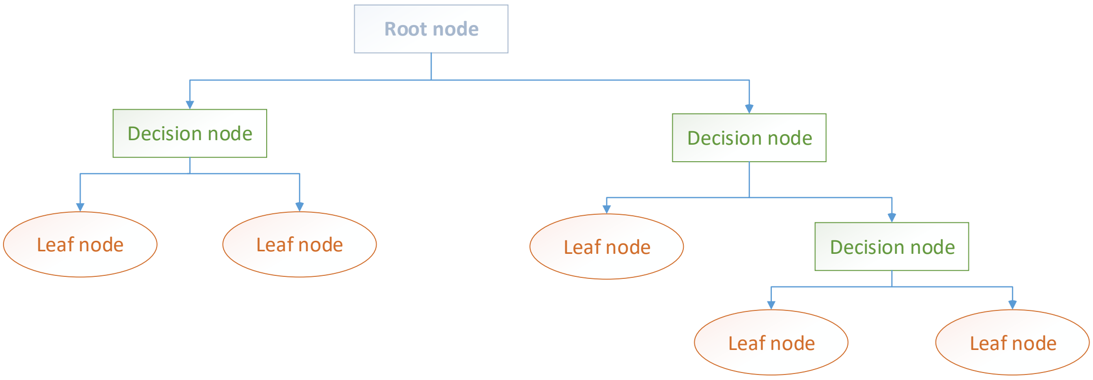


Fig. 8. The decision tree model

In a decision tree, the algorithm commences from the root of the tree. It evaluates the value of the data point using the condition in the root node. Based on this evaluation, the data point will follow a specific branch to the next decision node. In this decision node, the algorithm compares the data again and moves the point further. This process continues until the point reaches a leaf node of the tree.

An example of the decision tree for harassment problems can be shown in Fig. 9. By analyzing the number of calls and the average call duration that a phone number makes per day; network administrators can predict whether this phone number is malicious or not.

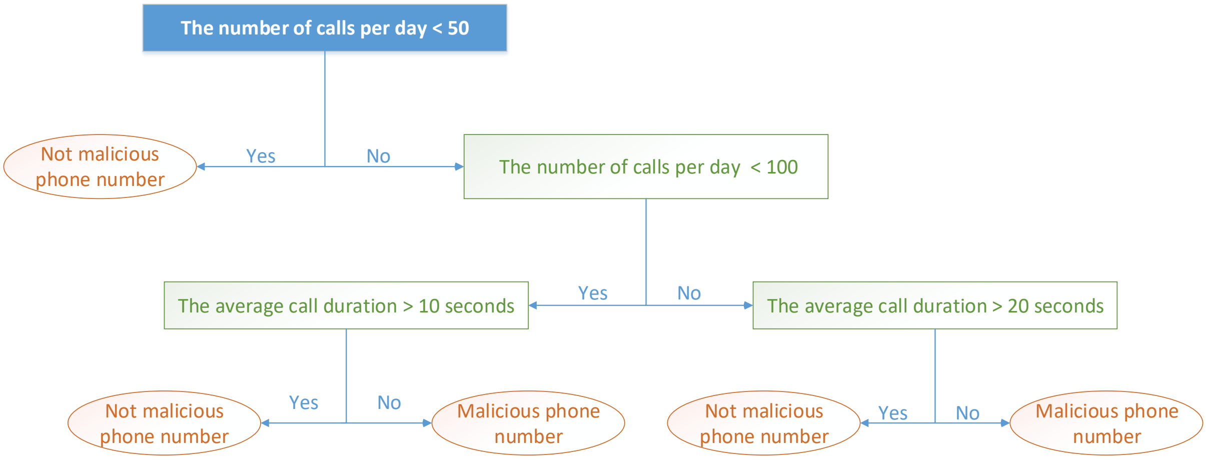

The number of calls per day < 50
Not malicious phone number
Yes
No
The number of calls per day < 100
The average call duration > 10 seconds
Yes
No
The average call duration > 20 seconds
Not malicious phone number
Yes
No
Malicious phone number
Not malicious phone number
Yes
No
Malicious phone number

Fig. 9. An example of the decision tree

4. EVALUATION OF CLASSIFICATION MODELS

In this section, the KNN algorithm, decision tree algorithm, and the neural network with logistic regression are evaluated with the task of predicting malicious telephone numbers. A simulated dataset is prepared for this evaluation. The dataset is just some examples to compare and evaluate the performance of the classification models. The 10-fold cross validation is also applied to utilize the datasets for better results [19]. The 10-fold cross validation divides the initial dataset into 10 folds, then it uses 9 folds to train and the last fold is used for validation. This training procedure repeats 10 times, so each fold becomes a test fold once.

It can be seen that the KNN algorithm has the best accuracy, while the decision tree algorithm's accuracy is similar to that of logistic regression. Fig. 10 shows a decision tree for the dataset with 60 instances, while Fig. 11 shows that of the dataset with 45 instances. In Fig. 10, the root node first checks whether the average call duration exceeds 15 seconds. If the average call duration of a phone number exceeds 15 seconds, it can be inferred that the user makes normal conversation calls (indicated by “no”). On the other hand, if the average call duration is below 15 seconds, the tree checks whether the number of calls per day exceeds 15. If the number of calls exceeds 15 calls, it can be predicted that the phone number is malicious (indicated by “yes”). If the number of calls is 15 or fewer, the phone number is predicted to be expected. In Fig. 11, the tree’s decision progress is similar.

Table 1: Comparison with the dataset of 60 instances

Number of instances in the dataset	Models	Accuracy
60 instances	Logistic regression	95%
	KNN (with K=3)	98.33%
	Decision tree	95%

Table 2: Comparison with the dataset of 45 instances

Number of instances in the dataset	Models	Accuracy
45 instances	Logistic regression	95.566%
	KNN (with K=3)	97.778%
	Decision tree	95.566%

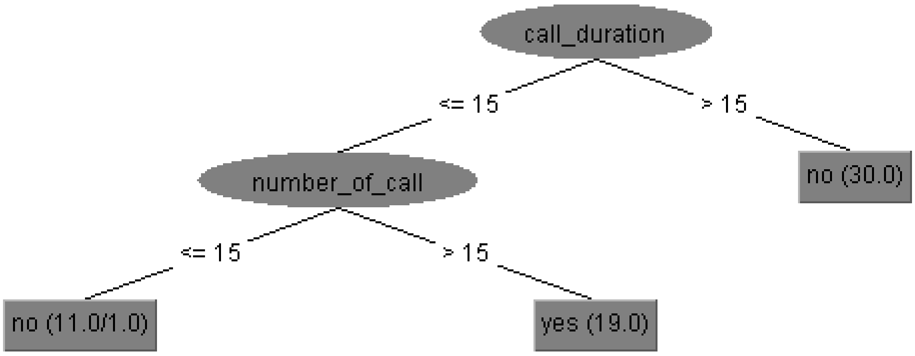


Fig. 10. The decision tree of the dataset with 60 instances

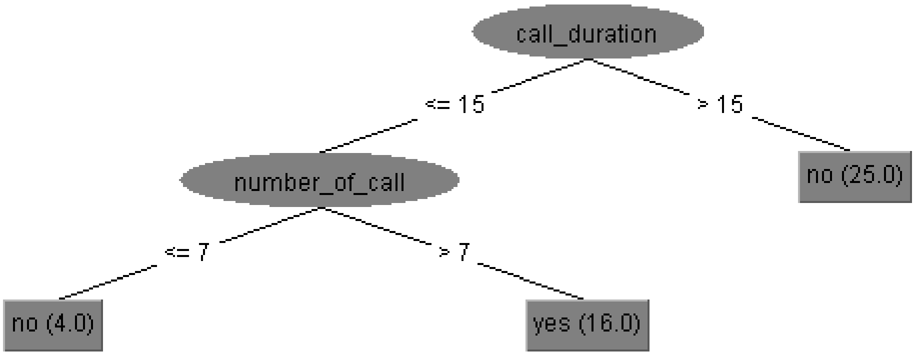


Fig. 11. The decision tree of the dataset with 45 instances

5. CONCLUSION

The malicious telephone numbers can harm the telecommunications network and disrupt the experience of mobile phone users. Machine learning classification algorithms can facilitate tracing these malicious telephone numbers. Network administrators can detect suspicious telephone numbers and impose restrictions on them thanks to these algorithms. In this manuscript, a simple comparison is conducted with the KNN algorithm, the Decision Tree algorithm, and the neural network with Logistic Regression.

REFERENCES

[1] Hicham, Magri & Abghour, Noreddine & Ouzzif, Mohammed. (2015). 4G System: Network Architecture and Performance.

[2] Zaame, I. & Mazri, Tomader & Elrhayour, A.. (2020). 5G: Architecture Overview And Deployments Scenarios. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XLIV-4/W3-2020. 435-440. 10.5194/isprs-archives-XLIV-4-W3-2020-435-2020. https://doi.org/10.5194/isprs-archives-XLIV-4-W3-2020-435-2020

[3] Tataria, Harsh & Shafi, Mansoor & Molisch, Andreas & Dohler, Mischa & Sjoland, Henrik & Tufvesson, Fredrik. (2021). 6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities. Proceedings of the IEEE. PP. 1-34. 10.1109/JPROC.2021.3061701. https://doi.org/10.1109/JPROC.2021.3061701

[4] Hayashi, Toshiki. (2012). Evolved Packet Core (EPC) network equipment for Long Term Evolution (LTE). Fujitsu scientific & technical journal. 48.

[5] ETSI TR 101 292, Public Switched Telephone Network (PSTN), 1999-09.

[6] Sauter, Martin. (2021). 5G New Radio (NR) and the 5G Core. https://doi.org/10.1002/9781119714712.ch6

[7] Sah, S. Machine Learning: A Review of Learning Types. Preprints 2020, 2020070230. https://doi.org/10.20944/preprints202007.0230.v1

[8] A. Bahaa, M. Shehata, S. M. Gasser, S. El-Mahallawy, "Call Failure Prediction in IP Multimedia Subsystem (IMS) Networks," in Applied Science Journal, 2022,12,8378. https://doi.org/10.3390/app12168378

[9] ETSI TR 122 115, Charging and Billing, 2000-01.

[10] C. Hsu, S. Wang, Y. Qiao, "Intrusion detection by machine learning for multimedia platform," in Multimedia Tools and Applications, 2021, pp. 29643-29656, https://doi.org/10.1007/s11042-021-11100-x

[11] A. R. Sammer, "A deep and machine learning comparative approach for networks intrusion detection", Asian Journal of Convergence in Technology, Vol 10 No.1 (2024), pp.98-103.

[12] R. P. Sinkar, "Use of Machine Learning Application for Business Perspective", Asian Journal of Convergence in Technology, Vol 10 No.1 (2024), pp.74-79.

[13] Dharmireddy, A., & Gottipalli, M. D. (2023). Social Networking Sites Fake Profiles Detection Using Machine Learning Techniques. Asian Journal For Convergence In Technology (AJCT) ISSN -2350-1146, 9(3), 09-15. https://doi.org/10.33130/AJCT.2023v09i03.002

[14] Kawli, D. P., Chaudhari, A. S., Ingale, P. D., Telange, G. A., & Banik, A. (2024). "Cryptocurrency Price Prediction Using Machine Learning", Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146, 10(1), 19-23. https://doi.org/10.33130/AJCT.2024v10i01.004

[15] Wegner, Sven. (2024). Neural Networks. 10.1007/978-3-662-69426-8_16. https://doi.org/10.1007/978-3-662-69426-8_16

[16] Geng, Yu & Li, Qin & Yang, Geng & Qiu, Wan. (2024). Logistic Regression. 10.1007/978-981-97-3954-7_4. https://doi.org/10.1007/978-981-97-3954-7_4

[17] Cunningham, Padraig & Delany, Sarah. (2007). k-Nearest neighbour classifiers. Mult Classif Syst. 54. 10.1145/3459665.

[18] Wang, Zijun & Gai, Keke. (2024). Decision Tree-Based Federated Learning: A Survey. Blockchains. 2. 40-60. 10.3390/blockchains2010003. https://doi.org/10.3390/blockchains2010003

[19] D. Anguita, Ghio A., S. Ridella, and D. Sterpi. K-fold cross validation for error rate estimate in support vector machines. In Proc. of the Int. Conf. on Data Mining, 2009.

[20] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

ASIAN JOURNAL OF ELECTRICAL AND ELECTRONIC ENGINEERING
Vol. 4 No. 2 2024
E-ISSN: 2785-8189

Improving Crowd Counting Performance: A Convolutional Neural Network Approach with Transfer Learning

Marwah M. Ahmeed[1] and Othman O. Khalifa[2]

[1]*Collage Of Electronic Technology, Bani Walid, Libya*
[2]*Libyan Center for Engineering Research and Information Technology, Bani Walid, Libya*

*Corresponding author: ookhalifa@gmail.com

(Received: 16 August 2024; Accepted: 27 August 2024)

Abstract— Precise crowd counting is critical to public safety and smart city planning since it solves the problems associated with the time-consuming manual counting of people in photos and videos. Transfer learning has become a key building block for improving crowd counting techniques, especially when used to Convolutional Neural Networks (CNNs). Because pretrained models already know the pertinent weights and architecture, using them in transfer learning minimizes computational demands and shortens training time. This paper presents a crowd counting method with an emphasis on optimizing the VGG16 model with a mall dataset. The results show that using VGG16 for transfer learning leads to higher performance when compared to more modern methods like AdaCrowd and PSSW models. In addition, the paper highlights how adaptable our proposed method is and how well it can transfer knowledge from one dataset to another.

Keywords: *Crowd Counting, VGG16, Enhanced Performance, Smart Cities, Public Safety, Convolutional Neural Networks (CNNs)*

1. INTRODUCTION

Crowd counting is a challenging task in computer vision that involves estimating the number of people present in each scene or image. It has a wide range of applications, such as traffic monitoring, crowd management, and security surveillance. The computer vision community has recently given crowd counting much attention, and several solutions have been put out to address this issue [1]. Traditional approaches to crowd counting involve manually designing features and using handcrafted algorithms for counting people [2]. However, these methods often suffer from low accuracy and scalability issues when dealing with complex scenes with many people. With the recent advances in deep learning, there has been a shift towards using deep neural networks for crowd counting [3]. The most advanced solution for this task is now deep learning-based, thanks to their impressive performance in crowd counting [4].

Deep neural networks require large amounts of labelled data to achieve high accuracy. However, collecting and annotating a large crowd counting dataset is a challenging and time-consuming task. Furthermore, the diversity and complexity of real-world crowd scenes make it difficult to capture and label all possible scenarios [5]. Therefore, transfer learning, a technique that enables the transfer of knowledge from one task to another, has become a popular approach in deep learning-based crowd counting [6].

Transfer learning can be used to leverage pre-trained deep neural networks that have been trained on large-scale datasets such as ImageNet [7]. The pre-trained models have already learned to recognize high-level features such as edges, shapes, and textures, which are also relevant to crowd counting. By fine-tuning the pre-trained models on a small crowd counting dataset, we can achieve high accuracy with limited labeled data [1]. This approach can significantly reduce the time and effort required for collecting and labeling crowd counting data [5]. In recent years, various pre-trained models have been proposed for transfer learning-based crowd counting. Among them, VGG16, a deep residual network with 16 layers, has shown promising results in several computer vision tasks.[8]. VGG16 has achieved state-of-the-art performance on the ImageNet dataset, and its deep architecture allows it to capture

DOI: https://doi.org/10.69955/ajoeee.2024.v4i2.63

high-level features in complex scenes [4]. Therefore, we propose to use VGG16 for transfer learning-based crowd counting in this research.

2. RELATED WORK

Crowd counting has become a crucial computer vision problem in recent years, with applications found in a variety of fields including public safety, event management, and surveillance. Scholars have investigated a range of approaches to improve the precision and effectiveness of crowd counting models. This section examines seminal research that addresses the problems related to crowd counting by utilizing Convolutional Neural Networks (CNNs) and transfer learning approaches. Early crowd counting research frequently used conventional computer vision methods. But the emergence of deep learning—and CNNs in particular—marked a paradigm change in this field. In a groundbreaking study, [5] presented the application of a multi-column CNN architecture for crowd counting.

This established the groundwork for later research to investigate CNNs' capacity to manage complicated scenarios with a range of pedestrian sizes. Khalifa et al. [15] explored transfer learning for crowd counting using ResNet50 on the Mall dataset. While transfer learning reduced the computational burden and training time, the results showed mediocre Mean Absolute Error (MAE) and Mean Squared Error (MSE) compared to other recent techniques. Further improvements are required to make this approach more beneficial. However, it's worth noting that not all transfer learning approaches yield the same results. Additionally, Feng et al. [9] proposed a new deep learning model called Spatiotemporal Convolutional LSTM (ConvLSTM) for crowd counting in videos. This model captures both spatial and temporal dependencies in crowd counting videos and has shown improved accuracy compared to traditional ConvLSTM models.

Table I. Summary of Related Work

Authors/ Year	**Methodology**	**Strengths**	**Limitations**
Yingying, *et al*, 2016 [18].	Multi-column CNN for single-image crowd counting (MCNN)	Effective for estimating crowd counts from single images	Limited to single images; may not capture temporal dynamics in videos
Deepak, *et al*, 2017 [10].	Switch-CNN	Improved accuracy through fine-tuning	Limited to specific architecture (VGG16)
Vishwana, et al, (2017 [11]	Contextual pyramid CNN for generating crowd density maps(CP-CNN)	High-quality crowd density maps; Improved accuracy	May require large-scale datasets for pre-training
Yuhong, et al, 2018 [12].	CSRNet with dilated CNNs for congested scenes	State-of-the-art accuracy and efficiency	May require extensive computational resources
Feng, et al, 2017 [9].	Spatiotemporal ConvLSTM for crowd counting in videos	Captures spatial and temporal dependencies; Improved accuracy	May be computationally intensive
Mahesh, et al, 2021 [17[.	AdaCrowd	Adapts crowd counting models to new, unlabelled target scenes using adversarial learning; Overcomes lack of labelled target data	Adversarial learning complexity; Network training complexity
Zhen, et al, 2020 [16].	PSSW	Accurate crowd counting with limited labelled data	Framework complexity; May require substantial computation
Khalifa, et al, 2022 [14].	Transfer learning with ResNet50 on Mall dataset	Reduced computational burden; Faster training	Mediocre performance compared to other techniques; Further improvements needed
Lijia Deng Yudong Zhang (2020), [13].	FOCNN	Superior performance in low-density scenarios	Limited applicability to specific crowd scenarios

Furthermore, Zhao et al. [16] proposed an active learning framework for crowd counting that combines several innovative components, including partition-based sample selection, density regression module, domain classification module, and Mix-up regularization. This framework enables accurate crowd counting with very limited labelled data.

Lastly, Reddy et al. [17] introduced AdaCrowd, a method that addresses the challenge of adapting crowd counting models to new, unlabeled target scenes. It leverages unlabeled target data during training and employs teacher-student learning framework combined with an adaptation module based on adversarial learning. These studies [9,14,16,18] provide valuable insights and methodologies for applying transfer learning in crowd counting tasks. They highlight the importance of leveraging pre-trained CNN architectures and fine-tuning them on crowd counting datasets to achieve improved accuracy while reducing training time and computational complexity.

Table 1 shows the summary of the methodology and advantages and disadvantages of published articles that are used as references for this literature review.

3. METHODOLOGY AND PROPOSED SOLUTION

In this work, a methodical strategy utilizing Convolutional Neural Networks (CNNs) and Transfer Learning is used to improve crowd counting performance. Figure 1 shows the proposed Solution.

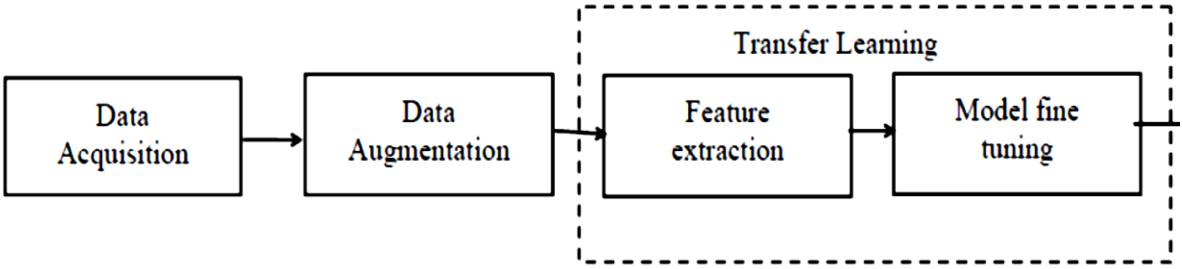


Fig. 1. Proposed Solution

A. *Dataset Acquisition*

The Mall dataset, a collection of surveillance images captured from a shopping mall, was obtained from https://paperswithcode.com/dataset/mall. The Mall dataset contains 2000 annotated images, and all images have a resolution of 320 x 240 each accompanied by a corresponding crowd count label. These labels accurately depict the number of individuals present within each image, ranging from a minimum of 11 people to a maximum of 53 people. The dataset's comprehensive annotations facilitate supervised learning, allowing us to train and evaluate crowd counting models effectively.

B. *Data Augmentation*

To enhance the model's generalization capabilities and reduce overfitting, various data augmentation techniques were employed, including Rotation, Scaling, and flipping. These techniques increase the diversity of the training data, enabling the model to learn robust features.

C. *Image* Preprocessing

Preprocessing the images is essential to optimize model performance. The following preprocessing techniques were applied to the images:

Resizing: All images were resized to a fixed dimension (e.g., 224x224 pixels) to ensure uniformity and compatibility with the VGG16 model architecture.

Rescale (Pixel Rescaling): This operation involves rescaling pixel values to a specific range, such as [0, 1]. It is typically expressed by the equation for each pixel.

If X is the original pixel value and X_rescaled is the rescaled pixel value: X_rescaled=X/255

ZCA Whitening: ZCA Whitening is a method that reduces data variance and enhances data quality. Feature-wise Standardization normalizes the distribution of features in the dataset, ensuring that each feature has a mean of zero and a standard deviation of one. Sample-wise Standardization normalizes data on a per-sample basis. It ensures that each sample in the dataset has a mean of zero and a standard deviation of one.

D. Model *Architecture* Design

The selection of an appropriate model architecture significantly impacts the crowd counting system's performance. In this paper, an enhanced VGG16 model, a deep convolutional neural network renowned for its effectiveness in computer vision tasks, was chosen as the backbone architecture. Figure 3 shows the VGG16 model after it was modified.

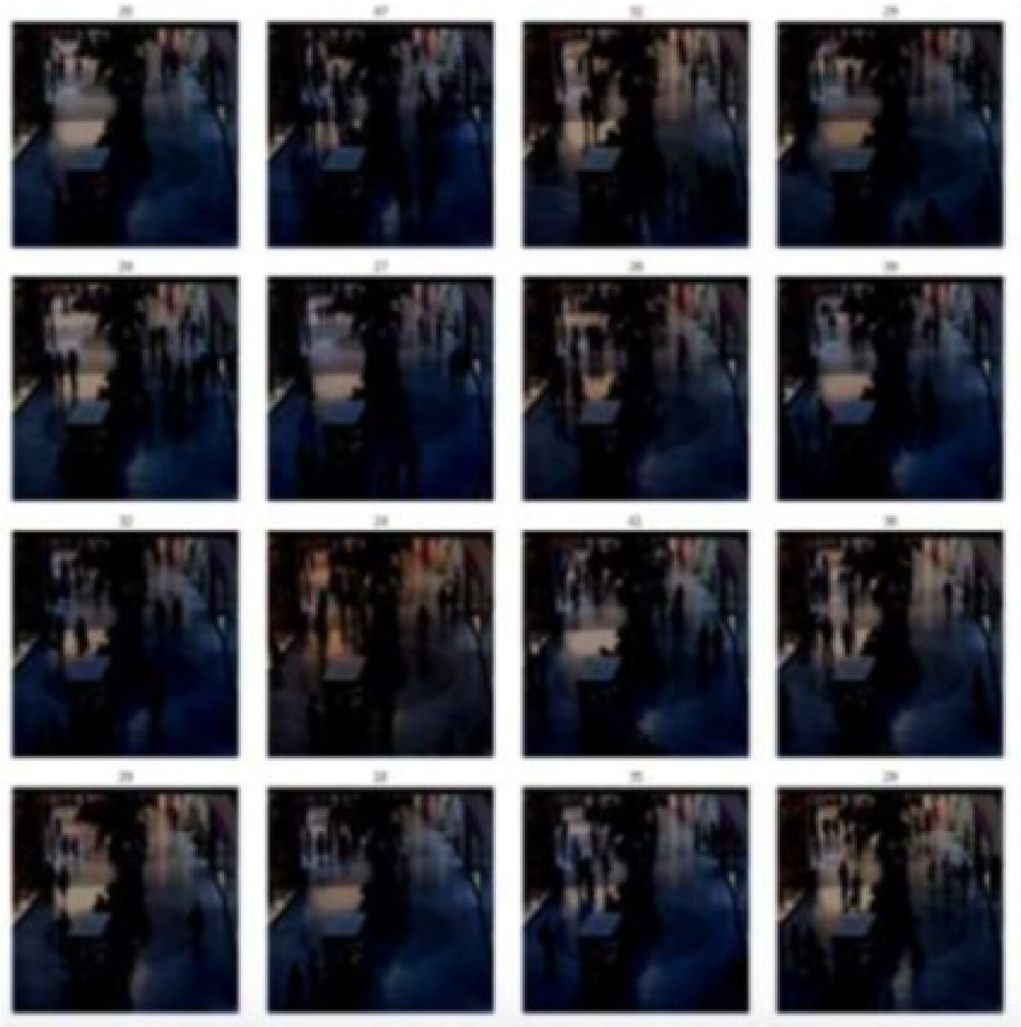


Fig. 2. Image Preprocessing view.

4. RESULTS ANALYSIS

To evaluate the performance of crowd counting models, mean absolute error (MAE) and mean squared error (MSE) are the most used parameters. MAE and MSE are defined as

$$MAE = \frac{1}{N}\sum_{i-1}^{N}|y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i-1}^{N}(y_i - \hat{y})^2$$

Where N is the number of test images
y_i is the number of actual people in the image and
y ̂ is the number of people estimated to be in the image

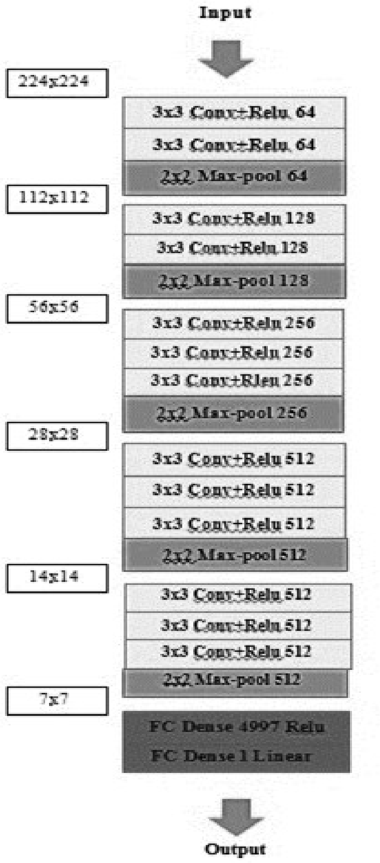


Fig. 3. Modified VGG-16 Architecture

4.1 Simulation Parameters Setup

A meticulous enumeration and elucidation of the parameters adopted are imperative. These parameters should be systematically presented in a table, accompanied by a clear elucidation of the type of Convolutional Neural Network (CNN) employed and its specific configurations. Table II shows the Optimized Parameters

Table II. Optimized Parameters

Parameter	**Range**
CNN Type	VGG16
Mini Batch Size	32
Optimizer	ADAM
Initial Learn Rate	0.001
Beta 1	0.8
Beta 2	0.95
Number Of Epochs	50
Number Of FC Layers	2

4.2 Finding and Outcome

The model is trained for 50 epochs and the model achieves 1.6 MAE, 4.1 MSE, 5.3 MAPE, 2.0 RMSE and 0.915 R^2. Figure 4 & 5 shows the deployment of the trained model in Smart City Dataset. However, the predicted count is more than the actual count.

5. COMPARATIVE PERFORMANCE ANALYSIS

The comparative performance analysis is shown in the table. Each method's performance is measured using metrics like Mean Absolute Error (MAE) and Mean Squared Error (MSE). Interestingly, the proposed method works better than the current methods, obtaining lower MSE and MAE values and indicating improved crowd density estimation accuracy. However, this paper offers insightful information about how well the strategy offered performs when compared to cutting-edge techniques from various years of publication.

Figure 6 shows the deployment of the trained model in the Beijing BRT dataset. However, the predicted count is less than the actual count.

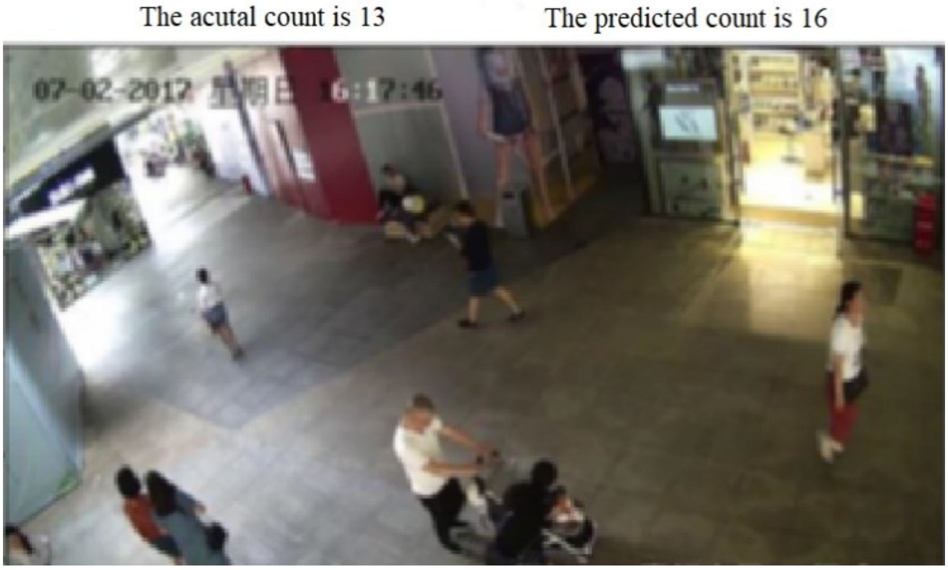


Fig. 4. An image Smart city dataset

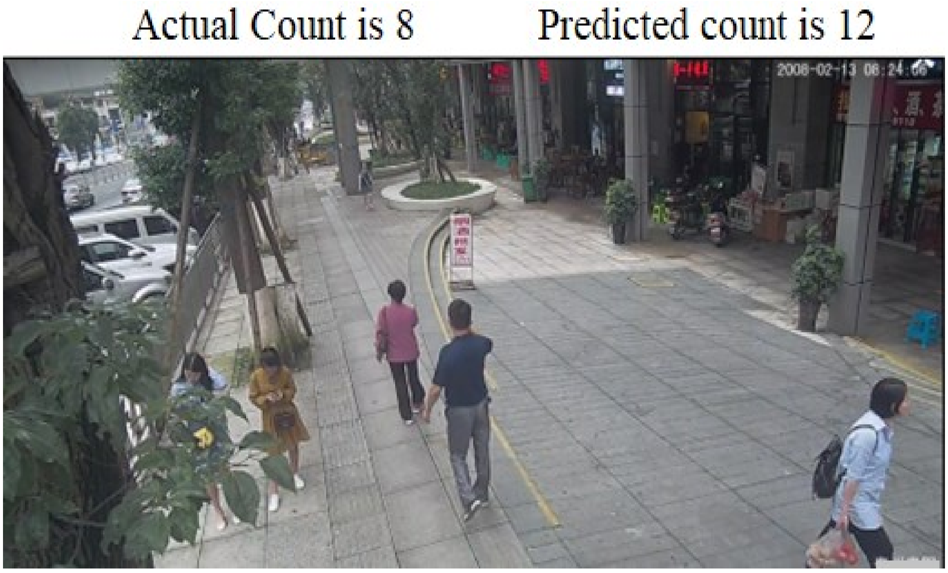


Fig. 5. An image Smart city dataset

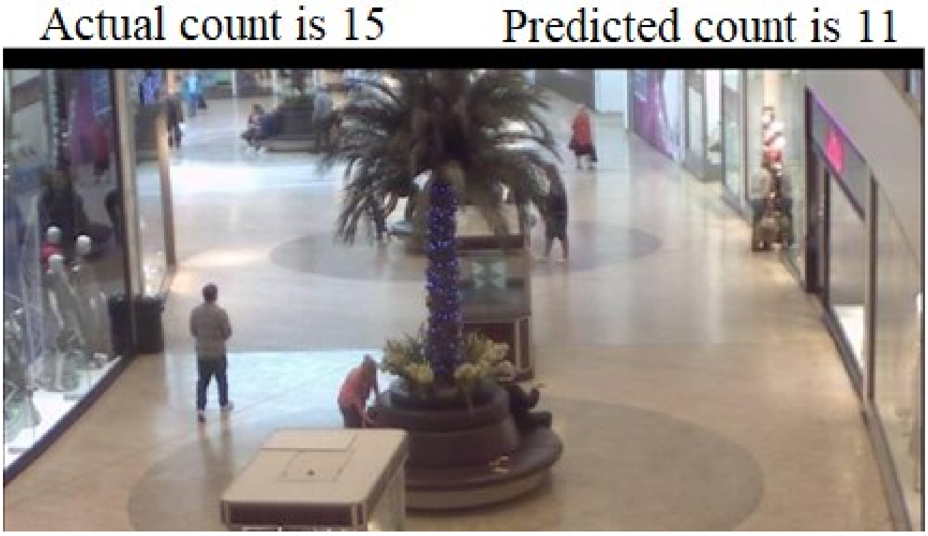


Fig. 6. An image of the Beijing BRT dataset

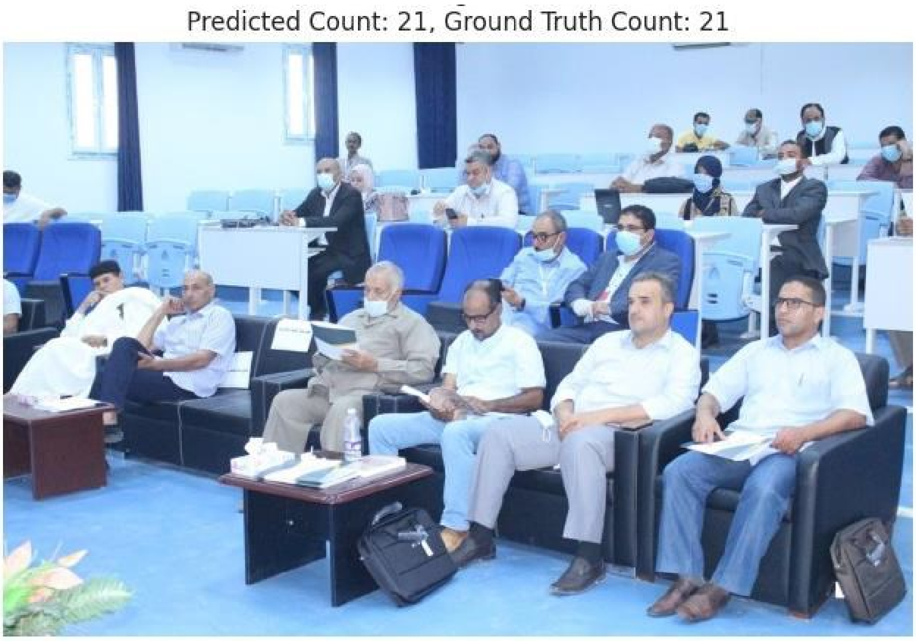


Fig. 7. An image from the College of Electronic Technology Bani Walid, Libya

Table III. Comparative Performance Analysis of Crowd Counting Methods

Methods	**MSE**	**MAE**	**YEAR**
AdaCrowd [17]	5	4	2021
MCCN [18]	8.5	2.24	2016
PSSW [16]	5.4	3.8	2020
Bidirectional ConvLSTM [9]	7.6	2.10	2017
ResNet50 with transfer learning [14]	15.5	3.31	2022
Proposed Method	**4.19**	**1.60**	**2023**

6. CONCLUSION

Crowd counting is a critical component of smart city planning and public safety. The conventional manual counting methods are resource-intensive, prompting the adoption of advanced techniques like transfer learning via Convolutional Neural Networks (CNNs) for crowd counting. This paper contributes to the field by presenting an innovative approach using the VGG16 model fine-tuned on a mall dataset. Transfer learning is known to be effective for problems involving crowd counting, providing significant computational and training time savings. Utilizing a pretrained model with predetermined weights and architecture, like VGG16, offers a strong basis for precise assessment of crowd density. The finding presented in this paper demonstrates the effectiveness of the proposed method and validate its efficacy.

REFERENCES

[1] C. Zhang, H. Li, X. Wang, and X. Yang, "Survey on crowd counting: Methods and datasets," Neurocomputing, vol. 399, pp. 67-89, 2020.

[2] A. B. Chan, Z. Q. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in European Conference on Computer Vision, 2008, pp. 412-425. https://doi.org/10.1109/CVPR.2008.4587569

[3] D. Wang, D. Zhang, Y. Chen, C. Zhang, and F. Yang, "Comprehensive study on convolutional neural network-based crowd counting methods," Neurocomputing, vol. 375, pp. 270-285, 2020.

[4] N. Liu, J. Zhang, K. Huang, and Z. He, "A Review on Deep Learning for Crowd Counting," IEEE Access, vol. 9, pp. 55801-55818, 2021.

[5] C. Zhang et al., "Data-driven crowd understanding: A baseline for a large-scale crowd dataset," IEEE Trans. Multimed., vol. 18, pp. 1048-1061, 2016. https://doi.org/10.1109/TMM.2016.2542585

[6] V. A. Sindagi and V. M. Patel, "A survey of recent advances in CNN-based single image crowd counting and density estimation," Pattern recognition letters, vol. 107, pp. 3-16, 2018. https://doi.org/10.1016/j.patrec.2017.07.007

[7] J. Pan, S. Liu, D. Sun, J. Yang, and C. C. Loy, "Crowd sampling the parameter space of deep neural networks for robustness," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11205-11214.

[8] J. Deng et al., "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition, 2009, pp. 248-255. https://doi.org/10.1109/CVPR.2009.5206848

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[10] X. Feng, X. Shi, and D. Yeung, "Spatiotemporal modeling for crowd counting in videos," in ICCV, 2017, pp. 5161-5169.

[11] A. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. https://doi.org/10.1109/CVPR.2017.429

[12] V. A. Sindagi and V. M. Patel, "Generating high-quality crowd density maps using contextual pyramid CNNs," in Proceedings of the IEEE International Conference on Computer Vision, 2017. https://doi.org/10.1109/ICCV.2017.206

[13] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. https://doi.org/10.1109/CVPR.2018.00120

[14] L. Deng, S. H. Wang, Y. D. Zhang, "Fully optimized convolutional neural network based on small-scale crowd," presented at the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), 2020. https://doi.org/10.1109/ISCAS45731.2020.9180823

[15] O. O. Khalifa, A. Albagul, A. H. Abdallah Hashim, N. Abdul Malik Hashim and K. N. Sakinahbt Wan Zainuddin, "Transfer Learning for Crowed Counting," 2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), Sabratha, Libya, 2022, pp. 248-253. https://doi.org/10.1109/MI-STA54861.2022.9837673

[16] X. Feng, X. Shi, and D. Yeung, "Spatiotemporal modeling for crowd counting in videos," in ICCV. IEEE, 2017, pp. 5161-5169.

[17] Z. Zhao et al., "Active crowd counting with limited supervision," presented at the ECCV 2020: 16th European Conference on Computer Vision, 2020. https://doi.org/10.1007/978-3-030-58565-5_34

[18] M. K. Krishna Reddy et al., "AdaCrowd: unlabeled scene adaptation for crowd counting," IEEE Transactions on Multimedia, vol. 24, pp. 1008-1019, 2022. https://doi.org/10.1109/TMM.2021.3062481

Ripeness Assessment and Quality Control of Mango Gold *Susu* using an E-Nose System

Nur Irdina Fakhrul Anwar, Nor F. Za'bah*, and Aliza Aini Md Ralib

Department of Electrical and Computer Engineering, Kulliyyah of Engineering, International Islamic University Malaysia, Kuala Lumpur

*Corresponding author: adah510@iium.edu.my

(Received: 27 August 2024; Accepted: 11 September 2024)

Abstract—In this paper, the development and implementation of an electronic nose (e-nose) system utilizing the MQ sensor series from MOS-type gas sensors to classify mango gold *susu* ripeness is presented. The system's performance was enhanced through machine learning techniques, including Principal Component Analysis (PCA) for data dimensionality reduction and Support Vector Machine (SVM) for classification. The SVM classifier demonstrated high accuracy, particularly in identifying unripe and overripe mangoes, with accuracy scores of 1.00 and 0.99, respectively. A comprehensive database of volatile organic compound (VOC) profiles was established, leading to a precise prediction model for assessing the different stages of ripeness based on the mango's VOC profile.

Keywords: *e-nose, VOC, Support Vector Machine, Principal Component Analysis*

1. INTRODUCTION

The fundamental solution to food waste is precise ripeness evaluation, which can prevent premature disposal and delayed consumption. Fruit ripeness has been determined manually by experienced farmers using fruit external characteristics such color, shape, firmness, and defect after post-harvest. As a result, only skilled and experienced farmers possess the ability to discern between different phases of fruit maturity using their own judgment. Diverse classifications for various fruits arise from the fact that professionals and farmers often evaluate ripeness classification in various manners based on their personal knowledge and experiences [1]. Consequently, human error may lead to inconsistencies in fruit maturity classification, impacting the quality control process. Therefore, there is a need for sophisticated, non-destructive alternatives since conventional quality control methods have limitations. These alternative methods offer less damage to fruits by utilizing characteristics such as aroma, color, and firmness for assessment. Various non-destructive techniques, such as NIR spectroscopy, electronic nose (e-nose), and RGB color sensors, have been explored to classify fruit ripeness stages accurately. Studies by Aghilinategh et al. [2] and Sabzi et al. [3] highlight the effectiveness of non-destructive methods like NIR spectroscopy and aerial video for fruit ripeness classification based on color. High accuracy levels, up to 97.88% for apple maturity classification, demonstrate the reliability of these techniques. Additionally, research by Baeitto and Wilson [4] emphasizes the role of fruit aroma in determining fruit quality and ripeness. E-nose instruments have shown promising results in accurately classifying ripeness stages for various fruits like blueberries, bananas, and apricots. The use of non-destructive methods, particularly e-nose technology, proves to be more effective, rapid, and non-damaging compared to traditional methods. This work will delve further into the usage of e-nose technology in monitoring the ripeness of mango gold susu, one of the common mangos in Malaysia. The main structure of an e-nose is illustrated in Fig. 1. Primarily, the e-nose systems are composed of a sensor array, a signal transducer, and a pattern recognition engine.

Fig. 1 shows that the sensor array is used to detect odor or in this work, the VOCs, generating and identifying the odor "fingerprint". However, unlike an olfactory system of mammals that has multiple receptor cells, the number of sensors for most electronic systems is limited.

DOI: https://doi.org/10.69955/ajoeee.2024.v4i2.65

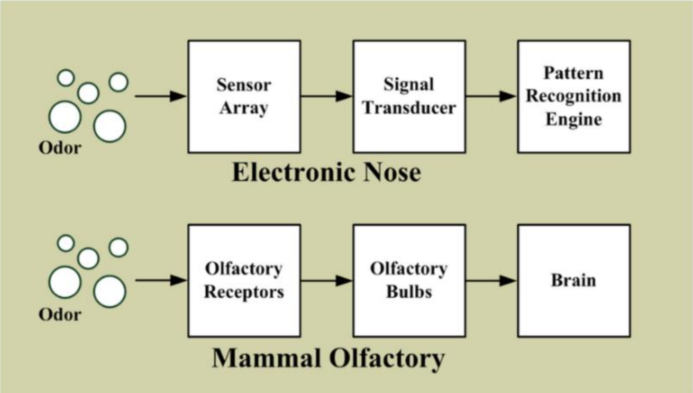


Fig. 1. An e-nose system [5]

2. METHODOLOGY

The methodology of this work involves a systematic four-step process, as shown in Fig. 2. In the first step, which is the data acquisition, the 6-MQ sensor array collects the VOC data within a controlled chamber to capture the volatile compounds emitted by the mangoes. The sensor, known for its versatility, cost-effectiveness, and sensitivity to various gases, is often used for gas detection in environmental monitoring and industrial settings. As mentioned before, the VOCs will go through the MQ series of sensors, which will detect the targeted gas and produce changes in its resistance. Table 1 shows the list of the 6 MQ sensors used in this work with the targeted gases.

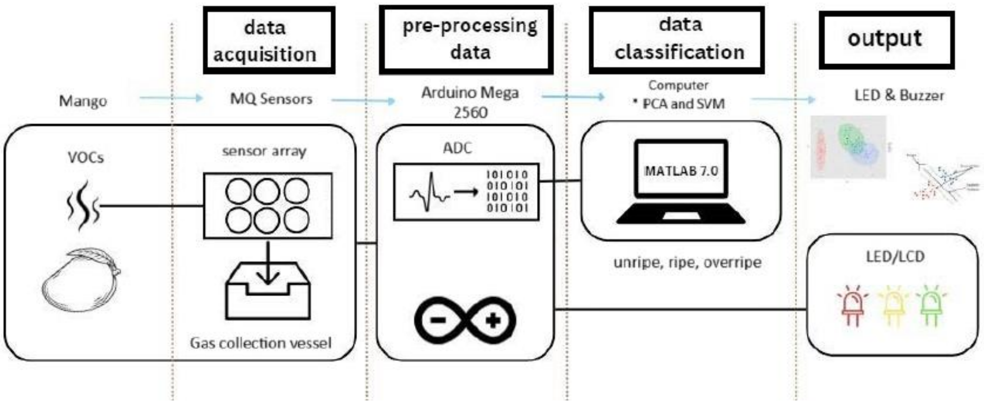


Fig. 2. The block diagram for the e-nose system for this work.

Table 1 - List of 6 MQ Sensors used for the e-nose system

Label	Sensor	Targeted Gases
S_1	MQ-2	Methane
		Butane
		LP Gas
		Smoke
S_2	MQ-3	Ethanol
		Alcohol
S_3	MQ-4	Methane
S_4	MQ-7	Carbon Monoxide
S_5	MQ-8	Hydrogen Gas
S_6	MQ-135	Ammonia
		Benzene
		Carbon Dioxide

This raw sensor data is then subjected to the pre-processing step, where an Arduino is employed for Analog-to-Digital Conversion (ADC). Additionally, baseline manipulation is performed on the raw e-nose data to normalize it, minimizing the impact of factors like temperature, humidity, and temporal drifts. Following pre-processing, the third step involves data classification utilizing Principal Component Analysis (PCA) and Support Vector Machines (SVM). PCA is employed to analyze data behavior and groupings, while SVM is used for training and classification. Finally, in the output step, this work employs three different colored LEDs, and an LCD connected to the Arduino, to produce the visual output where each color of the LED represents a specific ripeness category.

Fig. 3 shows the ripening stage of the mango fruit. Sample A's first batch of mangoes was in its early ripeness stage. After approximately seven days of storage, sample A became ripe, with a bright yellow color and a soft texture i.e. sample B. Then, overripe phases which took around five days, caused the mangoes to have a dark yellow color and a softer texture (sample C). Hence, we can safely say that the skin color and texture of the mangoes contributed to the assumption that they were at different stages of maturity. The machine learning model will categorize these samples by testing them with the e-nose setup.

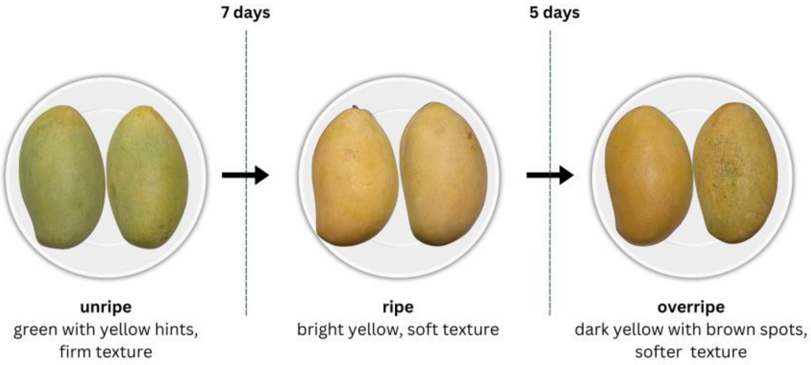


Fig. 3. The mango gold *susu* ripeness stages – unripe (sample A), ripe (sample B) and overripe (sample C)

2.1 Data Acquisition

On top of the array of sensors, the hardware configuration also consists of a smart digital thermometer and hygrometer for the recording of temperature and humidity. In order to calibrate the sensors for the first time, the sensors were preheated through electricity for 48 hours. After the calibration process, the MQ gas sensors must be preheated again for at least 30 minutes for the data collection to ensure the stability of the readings. This is based on the preliminary study that was carried out to guarantee that sensors' responses were stable during the data collection. This study is performed by comparing the sensor's reading obtained after 15 minutes against after 30 minutes. As shown in Fig. 4, it can be seen that notably, the sensors' readings were substantially more constant after 30 minutes.

Next, different types of VOCs produced by the mangoes during the three stages of ripening are collected. An airtight acrylic container is used in order to confine all the gas emitted from the mangoes. Three mangoes from each category were prepared for the data acquisition phase. Each mango was placed inside a container, and the sensors collected the information for 15 minutes, generating approximately 120 datasets per category.

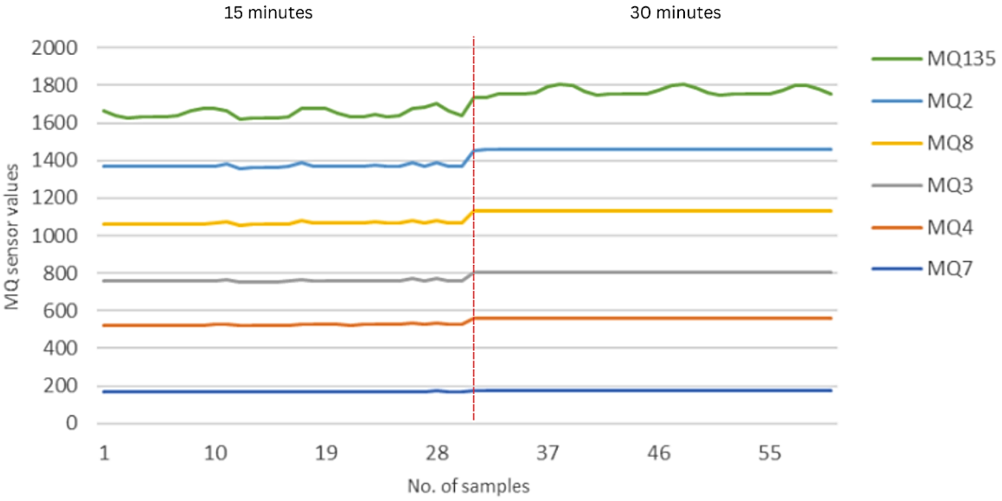


Fig. 4. MQ gas sensors' readings at the interval of 15 minutes vs 30 minutes

2.2 *Data Pre-processing and Data Classification*

The primary function of the Arduino microcontroller is to perform the Analog-to-Digital Conversion (ADC) procedure, which converts analog signals from the gas sensors into a digital format that can be classified. The constant analog signals generated by the gas sensors indicate the concentration of the gases they have detected. The Arduino code uses digital representations ranging from 0 to 1023 in a 10-bit ADC as the input data for the data classification stage.

A sample dataset's data analysis and classification are performed using PCA and SVM algorithms using PyCharm. PCA handles high-dimensional datasets and separates classes, especially in non-linear relationships. By reducing the dimensionality of the feature space while retaining essential information, PCA aids in uncovering the underlying patterns and structures in the MQ sensor readings. Fig. 5 illustrates the block diagram for the data analysis and classification. This work aims to use PCA for data grouping, while SVM is used for training and classification. PCA reduces the computational load on SVM during pre-processing, ensuring an efficient framework for precise ripeness evaluation based on the MQ sensor readings. The classification is based on three levels of mango ripeness: unripe, ripe, and overripe.

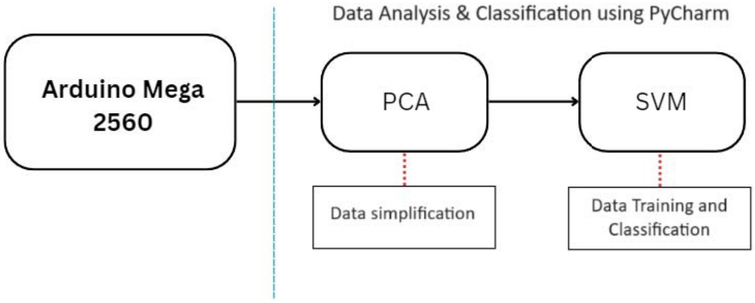

Data Analysis & Classification using PyCharm
Arduino Mega 2560
PCA
SVM
Data simplification
Data Training and Classification

Fig. 5. The block diagram for data analysis and classification

Several measures such as accuracy, precision, recall, and F1 score may be used to assess the SVM model's performance. However, in this work, the accuracy parameter is employed. The formula for the accuracy is as shown in Eq. (1) where:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

In this work, the definition of the accuracies are as follows:

- True Positives (TP): SVM correctly predicts positive classes, identifying some fruit as ripe when it is indeed ripe in ripeness assessment.
- True Negatives (TN): SVM correctly predicts negative classes, identifying some fruit as unripe or overripe when it is indeed unripe or overripe in ripeness assessment.
- False Positives (FP): SVM incorrectly predicts positive classes, mistakenly identifying an unripe or overripe fruit as ripe in ripeness assessment.
- False Negatives (FN): SVM incorrectly predicts negative classes, mistakenly identifying a ripe fruit as unripe or overripe in ripeness assessment.

3. RESULT AND DATA ANALYSIS

The illustration in Fig. 6 depicts significant distinctions in the sensors' responses corresponding to different ripeness stages. The data trends provide a solid foundation for the SVM model that allows it to classify the stages of ripeness.

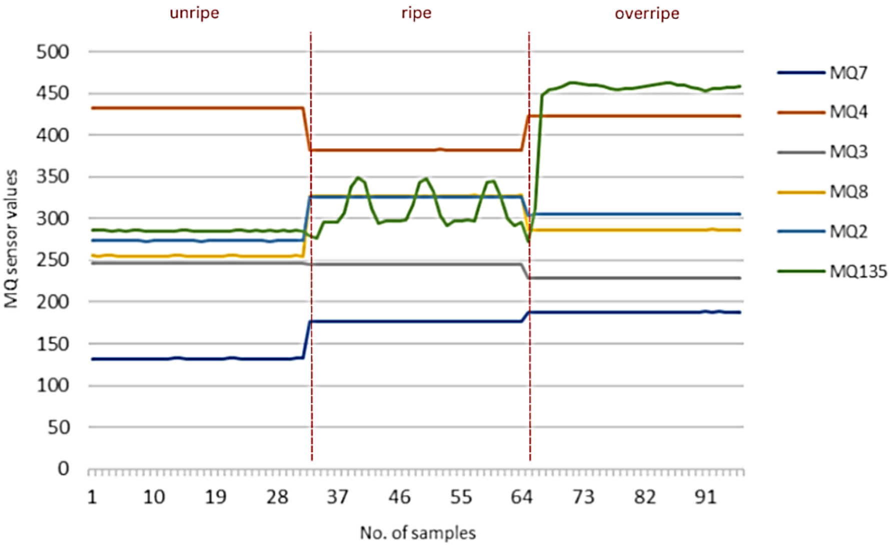
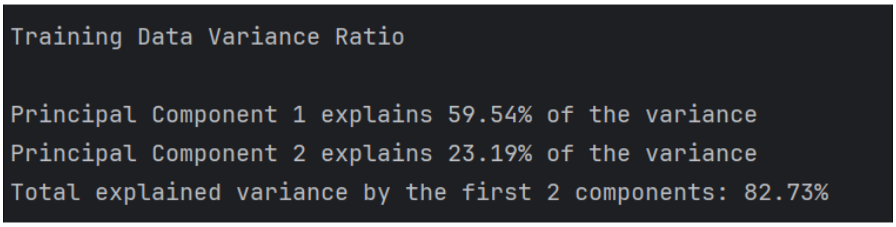


Fig. 6. Signal responses of the MQ gas sensors based on three stages of ripeness

3.1 *Principal Component Analysis (PCA)*

The output of the PCA analysis was promising, as shown in Fig. 7. The first principal component (PC1) explained a significant 59.54% of the variation in the sensor data. This indicates that the data has a dominant pattern or structure that PC1 is able to capture well in accordance with the three stages of ripeness. The second main component (PC2) accounted for an extra 23.19% of the variation, providing further information about the data's underlying structure. Significantly, by focusing simply on the first two main components, we were able to capture 82.73% of the overall data variability.

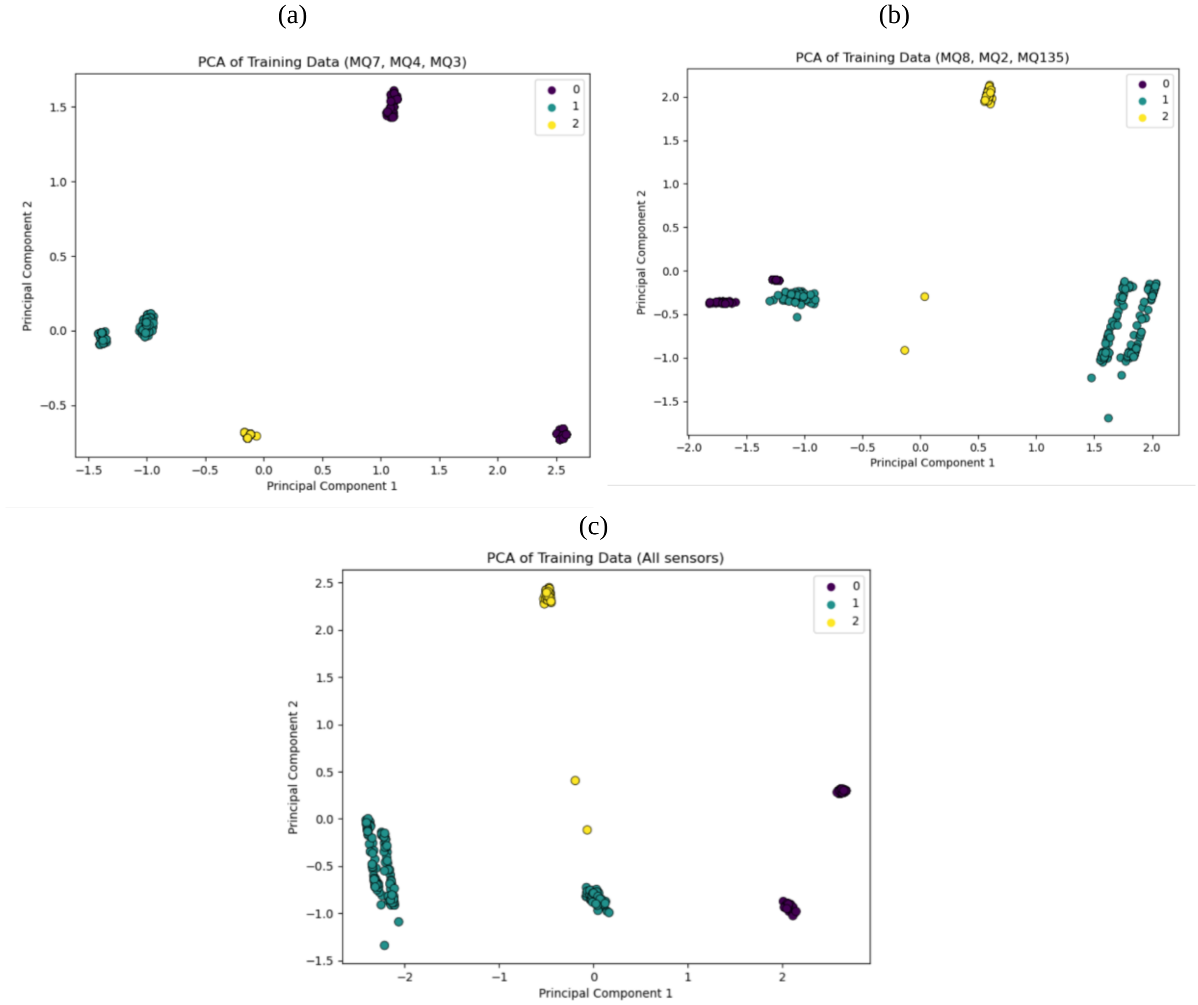


```
Training Data Variance Ratio

Principal Component 1 explains 59.54% of the variance
Principal Component 2 explains 23.19% of the variance
Total explained variance by the first 2 components: 82.73%
```

Fig. 7. Training Data Variance Ratio via PCA

The clustering patterns of the PCA model for mango ripeness classification are represented by a two-dimensional (2-D) scatter plot that shows individual mango samples as data points, with each color designated by numbers 0,1, and 2 representing different state of ripeness which are unripe, ripe, and overripe, respectively. This 2D plot reveals how the PCA model manages data based on the combined input gathered from particular MQ gas sensors. Data from the sensors are divided into three feature sets, namely:
Feature 1 which is the combination of MQ3, MQ4, and MQ7 (Fig. 8a)
Feature 2 which is the combination of MQ2, MQ8, and MQ135 (Fig. 8b)
Feature 3: which is the combination of all sensors. (Fig. 8c)


(a)

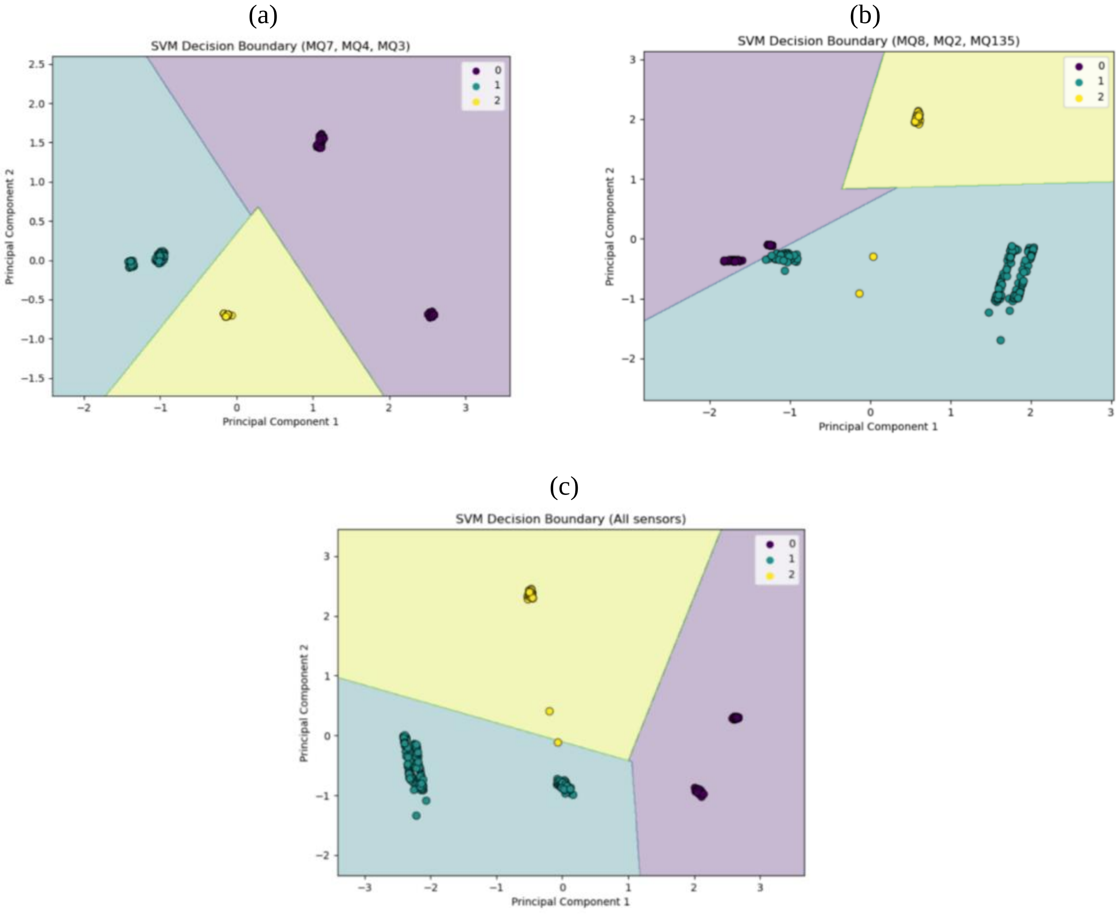
(b)

(c)

Fig. 8. 2-D graph plot of PCA clustering (a) Feature 1 (b) Feature 2 (c) Feature 3

3.2 *Support Vector Machine (SVM)*

Similar to the PCA clustering visualizations, the SVM model's decision boundaries for the ripeness classification are plotted in three independent 2-D graphs, as shown in Fig. 9. The decision boundaries created by the SVM model are reflected by the hyperplanes that divide the different ripeness classes based on the data collected from sensors.


(a)

(b)

(c)

Fig. 9. 2-D graph plot of SVM Decision Boundaries for (a) Feature 1, (b) Feature 2, (c) Feature 3

3.3 *Classification*

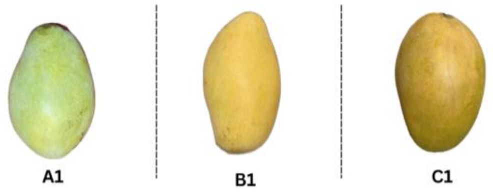
Using the testing data sets, the SVM model achieves a perfect accuracy of 100% in classifying the unripe category when utilizing all three features. This demonstrates that the model successfully learnt and incorporated the sensor data patterns to distinguish the unripe mangoes. However, both Feature 1 and Feature 2 failed to classify the ripe mango samples. This suggests that the sensor combinations did not offer enough information for the algorithm to correctly detect ripe mangoes. Nonetheless, the usage of Feature 3 resulted in an accuracy of 73% in detecting the ripe samples. Overall, as shown in Table 2, the accuracy of the e-nose system using the SVM model with Feature 3, which incorporates all six sensors, varies depending on the ripeness class. The system achieved accuracy rates of 100% for unripe, 73% for ripe, and 99% for overripe samples.

Table 2. The accuracy of SVM based on the three different sets

Ripeness	**Feature 1**	**Feature 2**	**Feature 3**
	MQ7, MQ4, MQ3	**MQ2, MQ8, and MQ135**	**All sensors**
Unripe	1.00	1.00	1.00
Ripe	0.00	0.00	0.73
Overripe	1.00	0.00	0.99

Three mango samples were used to test the mango ripeness classification using the Feature 3 model. As shown in Table 3, all three test samples—A1, B1, and C1—were accurately identified according to their respective ripeness stages. During the testing process, the sensor readings were recorded alongside the humidity and temperature values. The test was then repeated with another different set of three mango samples, yielding similar results.

Table 3. Datasets of the test samples



Sample	MQ2	MQ3	MQ4	MQ7	MQ8	MQ135	Humidity (%)	Temp (ºC)	Outcome
A1	270	215	408	143	217	307	92	30.0	Unripe
B1	267	205	362	170	241	319	89	29.2	Ripe
C1	266	194	366	161	258	316	89	29.4	Overripe

4. CONCLUSION

In this work, 6 MQ sensor series from the MOS-type gas sensor was utilized to build an e-nose system, which has been successfully implemented in classifying the mango gold *susu* ripeness stages. In addition, the system's performance in the classification phase using machine learning techniques by identify the VOCs combination for each stage has been demonstrated using PCA and SVM. In conclusion, a comprehensive database of VOC profiles for mangoes at different ripening stages has been compiled, and a predictive model to evaluate the mango ripeness based on these VOC profiles has been developed using statistical learning approaches. Based on the Feature 3 model, the e-nose system has demonstrated that it can accurately classify various stages of mango ripeness, achieving precision comparable to that reported in previous studies.

REFERENCES

[1] "Tackling food loss and waste: A triple win opportunity," Newsroom. https://www.fao.org/newsroom/detail/FAO-UNEP-agriculture-environment-food-loss-wasteday-2022/en

[2] N. Aghilinategh, M. J. Dalvand, and A. Anvar, "Detection of ripeness grades of berries using an electronic nose," Food Sci Nutr, vol. 8, no. 9, pp. 4919-4928, Sep. 2020, https://doi.org/10.1002/fsn3.1788

[3] M. F. Mavi, Z. Husin, R. Badlishah Ahmad, Y. M. Yacob, R. S. M. Farook, and W. K. Tan, "Mango ripeness classification system using hybrid technique," Indonesian Journal of Electrical Engineering and Computer Science, vol. 14, no. 2, pp. 859-868, May 2019, https://doi.org/10.11591/ijeecs.v14.i2.pp859-868

[4] M. Baietto and A. D. Wilson, "Electronic-nose applications for fruit identification, ripeness and quality grading," Sensors (Switzerland), vol. 15, no. 1. MDPI AG, pp. 899-931, Jan. 06, 2015. https://doi.org/10.3390/s150100899

[5] Chiu, Shih-Wen, and Kea-Tiong Tang. 2013. "Towards a Chemiresistive Sensor-Integrated Electronic Nose: A Review" Sensors 13, no. 10: 14214-14247. https://doi.org/10.3390/s131014214

A Comparative Study Between Transformer Coupled Un-Tuned and Tuned Power Derivative Circuits for Ultrasonic Cleaning System

S. M. A. Motakabber*, Md. Mominul Hoque, and A. H. M. Zahirul Alam

Dept. of Electrical & Computer Engineering, International Islamic University Malaysia,
Kuala Lumpur, Malaysia

*Corresponding author: amotakabber@iium.edu.my

(Received: 20 June 2024; Accepted: 25 September 2024)

Abstract— This paper discusses the challenges in developing a high-efficiency driver circuit for an ultrasonic cleaning system. The main issue lies in finding the suitable driver circuit and drive signal to ensure the correct frequency level for the ultrasonic transducer. This difficulty affects the removal of contaminants from inaccessible areas due to the inability to control cavitation bubbles. To address these problems, the authors studied the behaviour of piezoelectric transducers and designed driver and output power circuits. The LTspice XVII software was used to design and simulate various circuits, and it was found that trapezoidal pulses were the most suitable drive signals for avoiding power loss and achieving reasonable efficiency. The generated output frequency of 38.6 kHz provides sufficient energy for cleaning various small metallic items like jewellery, wristwatches, glasses, etc. The findings of this study will be helpful for those conducting research in this area in the future.

Keywords: *Ultrasonic, Piezoelectric Effect, Ultrasonic Transducer and Megasonic Transducer.*

1. INTRODUCTION

Ultrasonic cleaning systems use sound waves generated by converting mechanical energy into electrical energy. The frequency of these waves ranges from 20 kHz to 170 kHz and depends on the specific application. A typical system consists of a tank filled with a liquid solvent, cleaning items, and a driver circuit or ultrasonic vibration generator. The driver circuit drives a piezoelectric transducer, which produces the ultrasonic waves. The design of the circuit can vary, but the ultimate goal is to create cavitation bubbles, which are essential for removing contaminants from the items. A basic block diagram of an ultrasonic cleaning system is shown in Fig. 1.

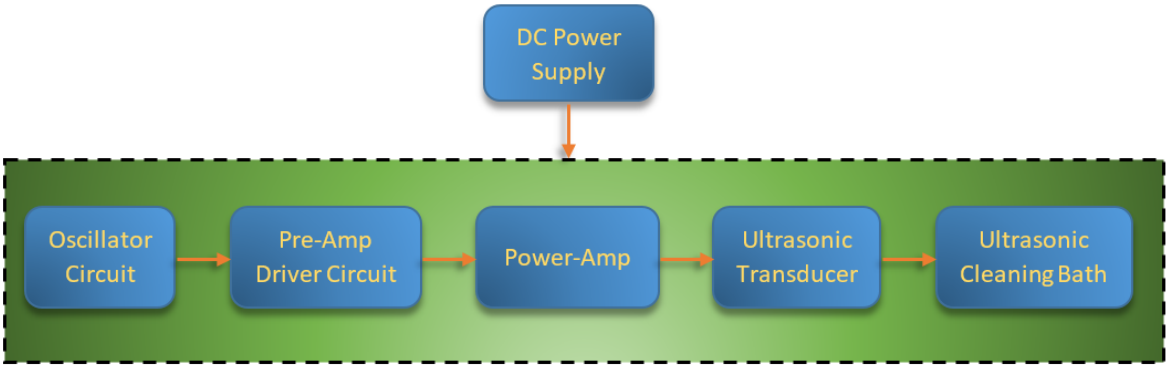


Fig. 1 Basic block diagram of an ultrasonic cleaning systems

In general, an electrical generator, an ultrasound transducer and a cleaning solvent tank are the fundamental

DOI: https://doi.org/10.69955/ajoeee.2024.v4i2.59

elements of an ultrasonic cleaning system. The leading tank designs in the business are heavy-duty, with a range of shapes and sizes ranging from 1 to 200 gallons, all in stainless steel. However, as time passed with technology evolution, many researchers in the field of electronics have performed studies to enhance the performance of ultrasonic cleaning, in which their testing methods focus on the nature of power circuits, controls and optimum frequency of activity for ultrasonic cleaning systems. The historical overview of ultrasonic cleaning traces its origins to the early 20th century. It highlights the pivotal role played by Paul Langevin's invention of the Langevin transducer in 1917, which laid the foundation for subsequent research and development in ultrasonic applications. The concept of sonochemistry, introduced by Wood and Loomis in 1927, further solidified the scientific understanding of ultrasonic cleaning and its underlying mechanisms. The emergence of the system can be seen to revive around the 1950s when several companies in the USA and UK started to develop and utilize it in their factory for unknown reasons [1].

In earlier systems, items were cleaned in a multi-tank process involving chlorinated solvents, pre-washing, and ultrasonic cleaning. Today, the cleaning process is typically streamlined, with contaminated items directly immersed in a solvent tank for complete cleaning, including cavitation. While the basic steps remain similar, advancements in solvents and operational frequencies have improved efficiency and effectiveness.

Traditional cleaning methods are used in various industries, including the heavy industry, food industry, medical instruments, clothing, and textiles. These methods often involve using hot solvents, detergents, or pressurized jets to remove contaminants [2]. However, many of these methods have limitations, such as environmental concerns associated with chlorinated solvents or the need for additional sterilization steps in medical applications.

The difference between ultrasonic and Megasonic cleaning is their frequency ranges. Ultrasonic cleaning typically operates between 25 and 270 kHz, while megasonic cleaning uses frequencies between 430 and 5 MHz [3]. It has a noticeable impact on the fluids. Random cavitation happens throughout the cleaning solution for ultrasonic washing since it requires lower frequencies, which later will automatically allow the cleaning force to reach all of the manually inaccessible areas. In comparison, the high frequencies of Megasonics induce fluid motions that contribute to stable cavitation without implosion, which differs from ultrasonic, where the cavitation is transient or cyclical. This implies that the cleaning happens in a line-of-sight manner, allowing only the side of the portion in front of the transducer to be cleaned [4]. The lower frequencies of ultrasonic cleaning are more suitable for general cleaning applications, while megasonic cleaning is often used for precision cleaning tasks, particularly in the electronics industry.

Ultrasonic cleaning applications are wide as they compromise the cleaning of dentures, 3D printing, textiles and many others. Even though higher frequency positively affects cleaning quality, the frequency of ultrasonic is not something to belittle. According to [5], the industrial ultrasonic cleaning frequencies are usually 20 kHz to 80 kHz, as frequencies beyond 100 kHz are suitable for precision cleaning. Megasonics, on the other hand, are particularly helpful in extracting sub-micron particles from flat surfaces. It is typically used mainly in the electronics industry to prepare silicon wafers at a relatively low risk of substrate impact. Fig. 2 shows the graphic comparison between ultrasonic cleaning and Megasonics cleaning.

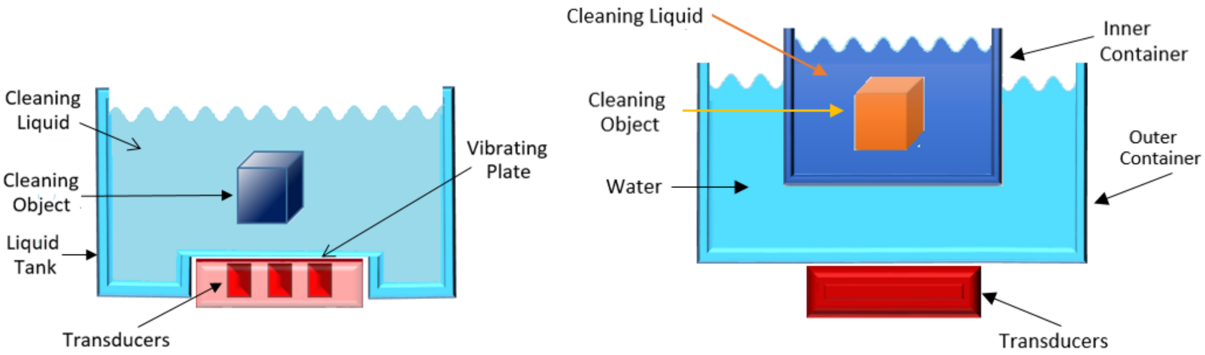


Fig. 2 The cross-section of (a) ultrasonic cleaning and (b) Megasonics cleaning [6]

An Ultrasonic Smart Cleaning Device (UCD) system proposed by Duran and Teke [7] that operates cleaning time independently to save resources and ensure healthy cleaning has been suggested. The system composition comprises four parts: the cleaning tank, ultrasonic transducer, inverter, and fluid analysis circuit. The conductivity and turbidity sensors are mounted with the container. These sensors submit data to the controller circuit encompassing two divisions, each with its microcontroller. The first microcontroller collects data from the sensors, such as the used liquid's temperature, conductivity, and turbidity. The algorithm tracks fluid changes and measures the cleaning period by regulating the fluid solution. Fig. 3 shows the block diagram of the overall system.

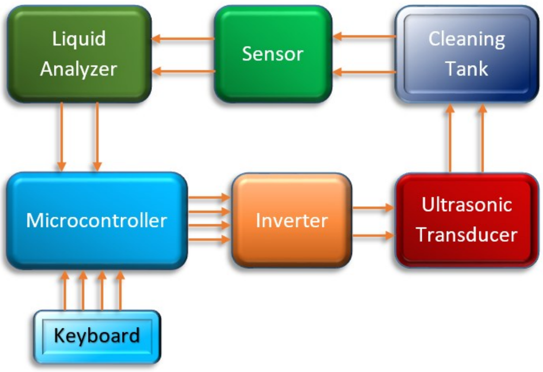


Fig. 3 Control system block diagram of UCD

The first section of the controller circuit involves an inverter system and an algorithm for temperature control. A full-bridging circuit of high frequencies was introduced to drive the ultrasound transducer for cavitation bubble generation. The microcontroller sets the operating frequency of the piezoelectric transducer to 38k Hz. The first microcontroller would then transfer data to another microcontroller to decide on the run period. The second microcontroller decides the cleaning process or works on the user interface. This overall process works dynamically with the help of software implementation.

Athira and Deepa presented a solar-powered ultrasonic cleaner with a twofold mode, charging and discharging [8]. The technique used is the multi-output half-bridge converter and a parallel-fed resonant inverter. The main objective here is to minimize part sizes so that they apply to low-power induction heating. Overall, this system can generate 230V voltage at 28 kHz. For load management purposes, the PI device is implemented. Lead acid batteries in converters are desirable for charging because they are cost-effective [9]. The proposed integrated system of this solar ultrasonic cleaning is shown in Fig. 4.

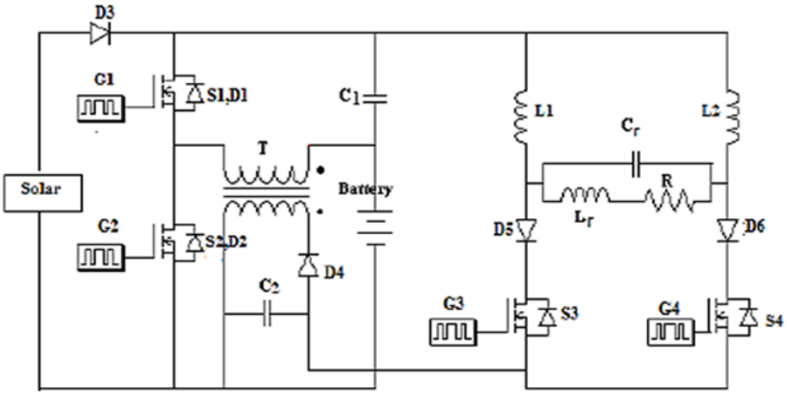


Fig. 4 Integrated system of solar-powered ultrasonic cleaner [9]

In [9], a Varying Frequency Ultrasonic Amplifier [10] offers the configuration of the amplifier with various frequencies. This amplifier system comprises transformers, bridge rectifiers, signal generators and a driver circuit consisting of a pre-amplifier and H-bridge. The signal generator, pre-amplifier and transducer are operated by a $220V_{AC}$ supply.

In [11], an incorporates redesigning the current Phase Controlled Thyristors (PCT) circuit to expand the operational range using impedance characteristics. In general, the system is composed of a few parts. First, the full-bridge rectifier converts the AC to DC voltage with a capacitor filter that filters the ripple voltage. A class-D inverter with two power switches as MOSFETs functions to convert back the DC voltage to AC voltage at a high switching frequency. While the series inductor extends the output voltage across the PCT load, signal conditioning is needed to detect output and input current, and finally, the ultrasonic cleaner and microcontroller. The configuration of the system can be seen in Fig. 5.

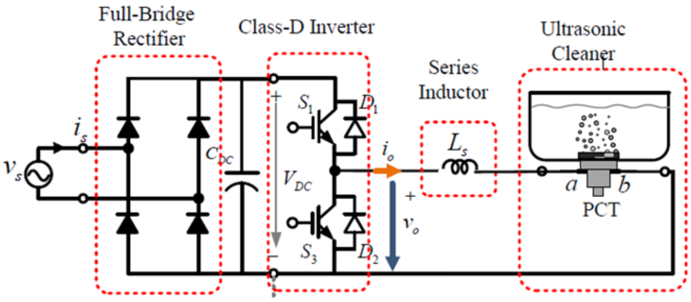


Fig. 5 A class-D inverter configuration system source [11]

2 OUTPUT POWER DRIVE CIRCUIT

A step-up transformer was used where the turns ratio (TR) of secondary to primary can be obtained from the transformer energy equations Eq. (1) to Eq. (3). For simplicity, the transformer was considered an ideal transformer. So, the amount of energy supplied to the transformer's primary is equivalent to the power delivered to the secondary of the transformer.

$$\frac{1}{2}(L_P)(I_P)^2 = \frac{1}{2}(L_S)(I_S)^2 \tag{1}$$

$$\sqrt{\frac{L_S}{L_P}} = \frac{I_P}{I_S} = \frac{V_S}{V_P} = \text{TR} \tag{2}$$

$$\text{TR} = \frac{N_2}{N_1} = \sqrt{\frac{L_S}{L_P}} \tag{3}$$

Where, L_P, I_P, V_P, N_P are the transformer's primary coil inductance, current, voltage, and number of trans coils, respectively. Similarly, L_S, I_S, V_S, N_S are the transformer secondary coil inductance, current, voltage and number of trans of the coil, respectively.

2.1 *H-Bridge Circuit and Piezoelectric Transducer*

The purpose of including an H-bridge circuit is to drive sufficient energy to the ultrasonic transducer. In the proposed circuit, MOSFETs of type IRFZ44N have been used due to their characteristics, which are low in resistance and can drive more power. The parameters and the design model of the piezoelectric transducer used the values of parallel capacitance, $C_d = 2430pF$, series inducatance $L_s = 110.83mH$, series resistance $R_s = 333.7\Omega$ and series capacitance $C_s = 153.4pF$ respectively.

3 OUTPUT POWER DRIVE CIRCUIT

Two types of arrangements were used for the transformer drives output circuit, namely untuned and tuned transformer circuits. To further understand the differences between these two circuits, the block diagram of both the untuned step-up transformer circuit and tuned secondary step-up transformer circuit can be seen in Fig. 6(a) and Fig. 6(b), respectively, where the red boxes and remarks indicate parts that have undergo iteration.

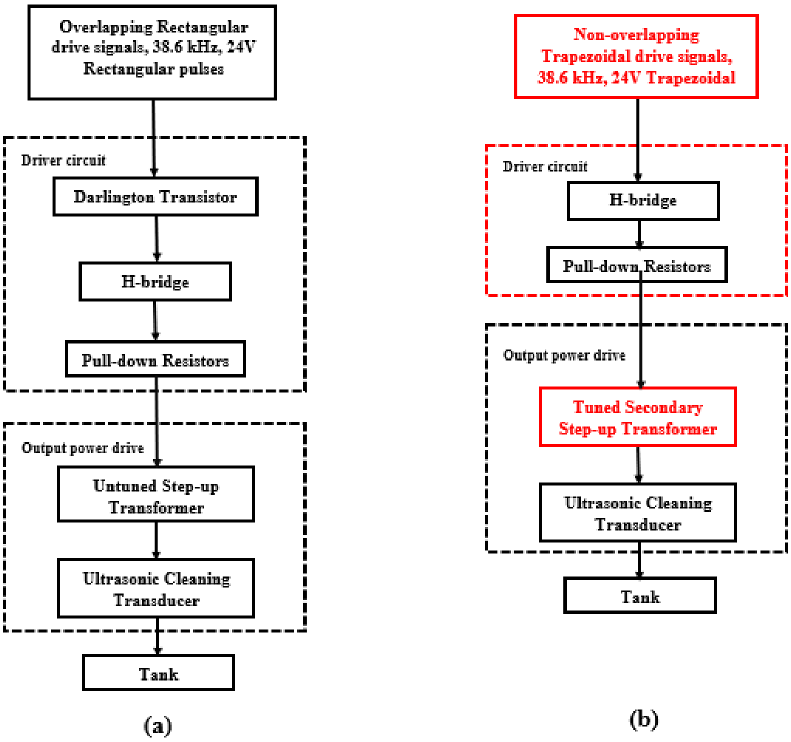

Overlapping Rectangular drive signals, 38.6 kHz, 24V Rectangular pulses
Driver circuit
Darlington Transistor
H-bridge
Pull-down Resistors
Output power drive
Untuned Step-up Transformer
Ultrasonic Cleaning Transducer
Tank
(a)
Non-overlapping Trapezoidal drive signals, 38.6 kHz, 24V Trapezoidal
Driver circuit
H-bridge
Pull-down Resistors
Output power drive
Tuned Secondary Step-up Transformer
Ultrasonic Cleaning Transducer
Tank
(b)

Fig. 6 Block diagram of the transformer circuit (a) untuned and (b) tuned secondary

3.1 Untuned Step-Up Transformer Circuit

The circuit shown in Fig. 7 is the overall circuit to be adapted in the ultrasonic cleaning system, where it comprises two primary sections: the driving circuit and the output power drive circuit. Areas outside the dotted red box indicate the driving circuit part, which includes a Darlington transistor, a pair of pull-down resistors, an H-bridge circuit containing four MOSFETs and also two voltage sources that drive them, which are a signal voltage drive and a power supply voltage. Meanwhile, areas included inside the dotted red box are the output power drive circuit that consists of a step-up transformer and an ultrasonic transducer. If a prototype is built, this part will be connected to the cleaning tank.

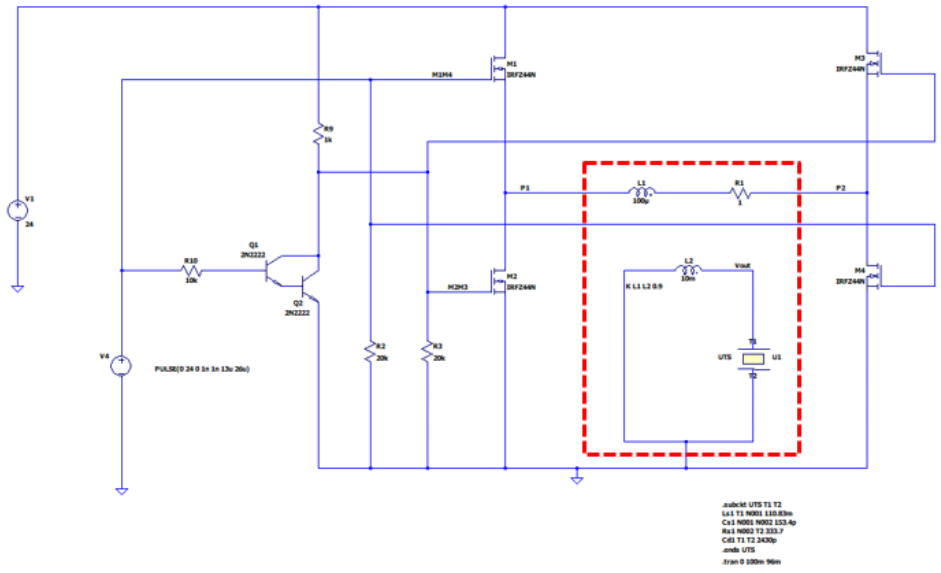
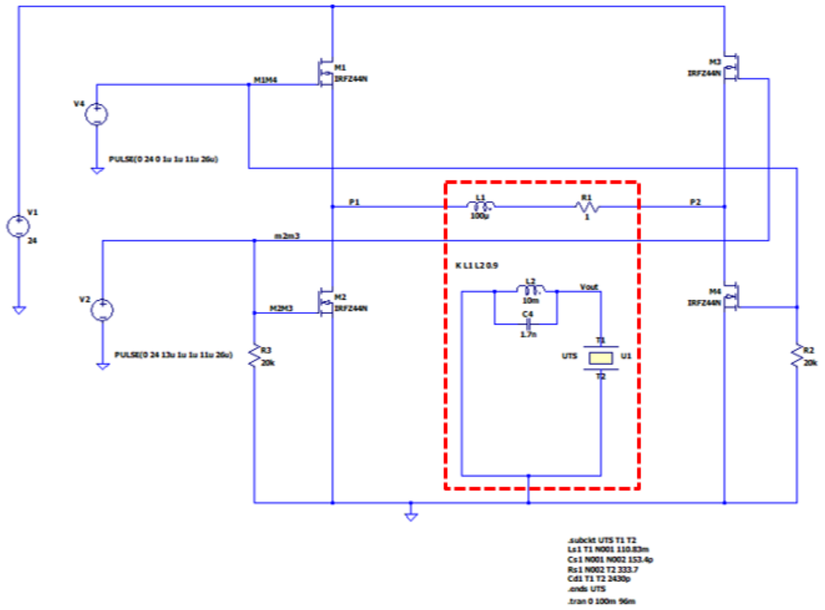


Fig. 7 Untuned step-up transformer circuit

3.2 Tuned Secondary Step-Up Transformer Circuit

The circuit shown in Fig. 8 results from iterations after a few changes on the driving circuit and ultrasonic cleaning system sections have been made on the previous circuit. On the driving circuit part, no changes were made to the power output except for the Darlington transistors, which were removed with their load and base resistors. Meanwhile, the secondary coil of the step-up transformer in the ultrasonic cleaning portion was tuned by adding a parallel capacitor. The idea was to isolate the inductances of the secondary from interfering with the transducer load and the calculation of the parallel capacitor. It can be seen that the areas outside the red box are the driver circuit, which is comprised of an H-bridge circuit and pull-down resistors. In contrast, the dotted red box includes the output power drive circuit comprising a tuned secondary step-up transformer and transducer.



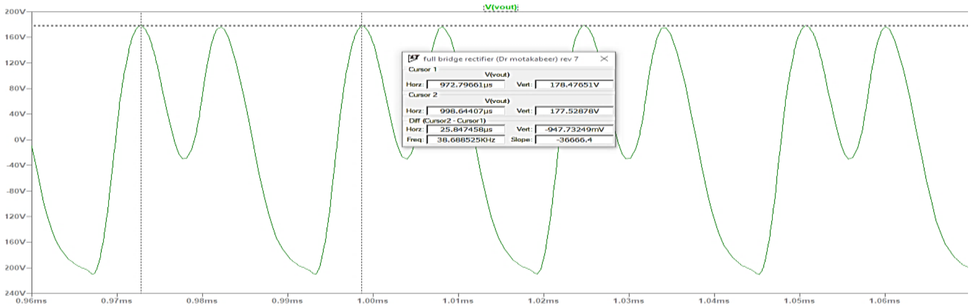
```
.subckt UTS T1 T2
Ls1 T1 N001 110.83m
Cs1 N001 N002 153.4p
Rs1 N002 T2 333.7
Cd1 T1 T2 2430p
.ends UTS
.tran 0 100m 96m
```

Fig. 8 Equivalent circuit of tuned secondary step-up transformer circuit

4 RESULT

4.1 Untuned Step-up Transformer Circuit

In order to compute the efficiency of the circuit, the power loss occurring at R1 and MOSFETs were recorded where the drain and source currents of MOSFETs during switching transitions were the power lost dissipated across Rd_{on}. Fig. 9 shows the output waveform of the driving transformer. The peak current through the drain and source of MOSFETs were obtained, as shown in Fig. 10(b). Also, for efficiency calculation, the current through the load was also recorded to compute the power dissipated across the resistive component of the transducer, Rs. In Fig. 10(c), it can be observed that an overlap switching occurs at the voltage of V(m2m3) and V(m1m4). It was assumed the pulse was triangular. Even though, based on Fig. 10, the output frequency of the system is recorded to be at 38.6kHz, based on the calculations, the efficiency rate of the circuit is 35.10%, which is quite unsatisfactory to be adapted to real-life adaptation.



full bridge rectifier (Dr motakabeer) rev 7

Cursor 1
V(vout)
Horz: 972.79661µs Vert: 178.47651V

Cursor 2
V(vout)
Horz: 998.64407µs Vert: 177.52878V

Diff (Cursor2 - Cursor1)
Horz: 25.847458µs Vert: -947.73249mV
Freq: 38.688525KHz Slope: -36666.4

Fig. 9: The oscillation frequency waveform of the system

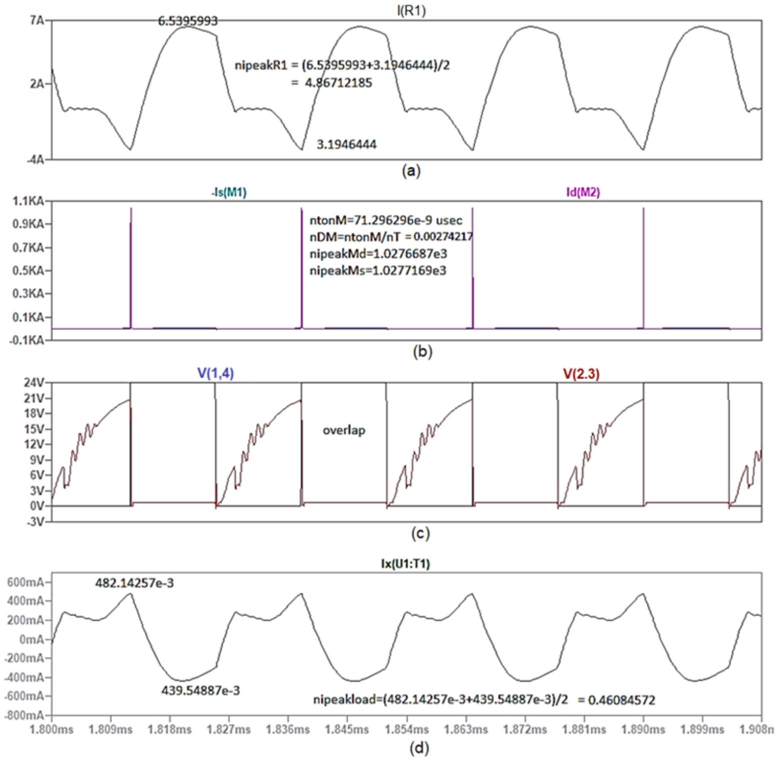
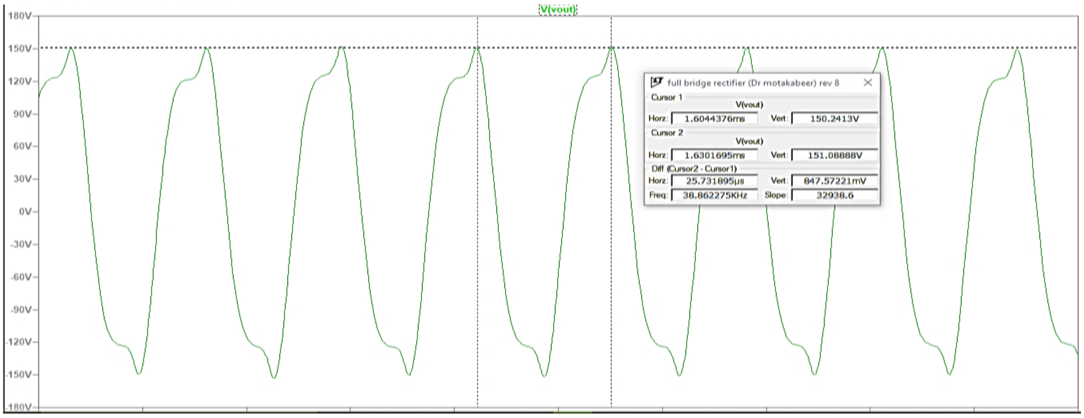


Fig. 10: The output waveforms for un-tuned transformer circuit, (a) current in R1, (b) current in MOSFET, (c) voltage at V(1,4) and V(2,3) and (d) current at the load

4.2 Tuned Secondary Step-up Transformer Circuit

The output waveforms generated from the tuned secondary step-up transformer circuit are shown in Fig 11, where four different readings have been analyzed. The technique previously used in the un-tuned step-up transformer circuit has also been adapted to compute the circuit's efficiency. The power loss occurring at R1 and MOSFETs were recorded where the drain and source currents of MOSFETs during switching transitions where the power lost dissipated across Rd_{on}. The peak current through the drain and source of MOSFETs were obtained, as shown in Fig. 12(b). In Fig. 12(c), it is observed that the trapezoidal signal has eliminated the overlapping scenario that once happened at the voltage of V(1,4) and V(2,3) in the previous circuit. Based on the calculations, it can be seen that this tuned secondary step-up transformer circuit was established operational at a reasonable efficiency of 70.51%. Also, based on Fig. 12, it has an output frequency of approximately 38.6 kHz, thus making it a better option to be adapted in the real-life ultrasonic cleaning system. A comparative analysis between the two circuits can be made based on a few significant parameters summarized in Table 1.



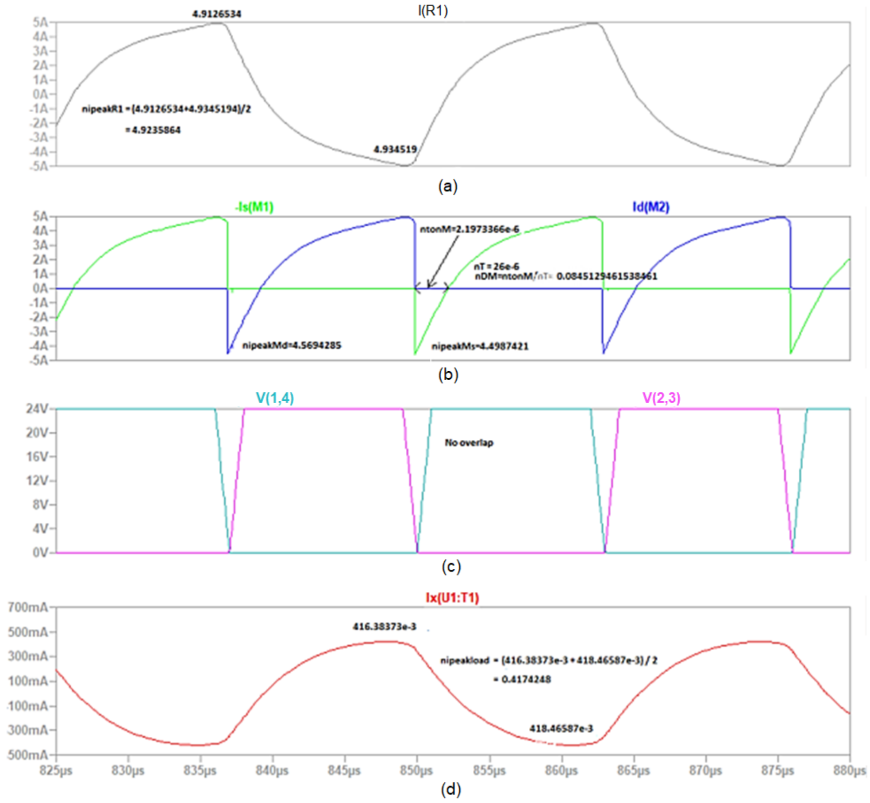
full bridge rectifier (Dr motakabeer) rev 8

Cursor 1
V(vout)
Horz: 1.6044376ms Vert: 150.2413V

Cursor 2
V(vout)
Horz: 1.6301695ms Vert: 151.08888V

Diff (Cursor2 - Cursor1)
Horz: 25.731895µs Vert: 847.57221mV
Freq: 38.862275KHz Slope: 32938.6

Fig. 11 The output frequency of the Tuned Secondary Step-up Transformer circuit



(a)

(b)

(c)

(d)

Fig. 12 The output waveforms for tuned transformer circuit, (a) current in R1, (b) current in MOSFET, (c) voltage at V(1,4) and V(2,3) and (d) current at load

Table 1: Comparisons between Untuned and Tuned step-up transformer circuits performance

Parameter	*Step-up Transformer Secondary Circuit*	
	Untuned	*Tuned*
Conversion power loss, P_{lost}	*65.519W*	*12.153 W*
Output power, $P_{rms}(Load)$	*35.10 W*	*29.068 W*
Conversion power Efficiency	*35.10%*	*70.52%*
Output current, $I_{rms}(Load)$	*0.326 A*	*0.295 A*
Output Voltage, V_{rms} $(Load)$	*108.742V*	*98.489V*

4 CONCLUSION

A comparative analysis between the two circuits can be made based on a few significant parameters summarized in Table 1. First and foremost, the most distinctive observation is that P_{lost} obtained in the untuned step-up transformer circuit is much higher compared to the P_{lost} Obtained in the other circuit. Even though the output power of the load in these two circuits has not much difference, due to the significant margin distinction in the value of P_{lost}. It has affected the circuit's performance, and this assessment was made using the efficiency computation made at both circuits. The high rate of P_{lost} in the untuned circuit caused the circuit to have an efficiency rate of 35.10%; meanwhile, the low value of P_{lost} in the tuned circuit has incremented its efficiency rate to 70.52%. Both circuits have the approximate output power value range of 30–35 W, suitable for producing sufficient cavitation bubbles during cleaning.

REFERENCES

[1] Grieser et al., Sonochemistry and the Acoustic Bubbles. Elsevier, 2015, pp. 1-9.

[2] M. Stanga, Ultrasound Cleaning in Sanitation, Wiley-VCH Verlag GmbH & Co. KGaA, 2010, pp. 473-476.

[3] A. A. Busnaina and G. W. Gale, "Ultrasonic and Megasonic Particle Removal," Proc. Precision Cleaning, pp. 347-360, 1995.

[4] B. Kanegsberg and E. Kanegsberg, Handbook for Critical Cleaning; Applications, Processes, and Controls, 2nd ed. Boca Raton: CRC Press, an imprint of Taylor & Francis Group, 2011. https://doi.org/10.1201/b10858

[5] M. Cai, S. Zhao, and H. Liang, "Mechanisms for the enhancement of ultrafiltration and membrane cleaning by different ultrasonic frequencies," Desalination, vol. 263, no. 1-3, pp. 133-138, 2010. https://doi.org/10.1016/j.desal.2010.06.049

[6] R. Magarajan, S. Awad, and K. R. Gopi, "Megasonic Cleaning," Developments in Surface Contamination and Cleaning, pp. 31-62, 2011. https://doi.org/10.1016/B978-1-4377-7885-4.10002-8

[7] F. Duran and M. Teke, "Design and Implementation of an Intelligent Ultrasonic Cleaning Device," Intelligent Automation and Soft Computing, pp. 1-10, 2018. https://doi.org/10.31209/2018.11006161

[8] S. Athira and K. Deepa, "Solar powered ultrasonic cleaner," 2014 Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD), pp. 1-6, 2014. https://doi.org/10.1109/AICERA.2014.6908245

[9] J. Marshal and K. Deepa, "Hybrid renewable energy system: Optimum design, control and maximum utilization with SIBB converter using DSP controller," 2014 Power and Energy Systems: Towards Sustainable Energy, 2014.

[10] K. M. C. Basa, K. P. S. Gomez, F. B. Navarro-Tantoco, A. S. Quinio, G. P. Arada, and C. B. Co, "Design of a varying ultrasonic frequency amplifier," TENCON 2012 IEEE Region 10 Conference, pp. 1-6, 2012. https://doi.org/10.1109/TENCON.2012.6412169

[11] J. Jittakort, J. Nimsontorn, B. Sirboonrueng, S. Chua-On, P. Pinpathomrat, and S. Chudjuarjeen, "A Class D Voltage Source Resonant Inverter for Ultrasonic Cleaning Application," 2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST). https://doi.org/10.1109/ICEAST.2018.8434484

Securing the IoT Edge Devices Using Advanced Digital Technologies

Abdul Manan Sheikh[1*], Md Rafiqul Islam[1], and Mohamed Hadi Habaebi[1],
Adnan Kabbani[2], Suriza Ahmad Zabidi[1], and Athaur Rahman bin Najeeb[1]
[1]*Dept. of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia*
[2]*Dept. of Electronics & Communication Engineering, A'Sharqiyah University, Ibra, Oman*

*Corresponding author: abdul.manan@asu.edu.om

(Received: 30 September 2024; Accepted: 2 October 2024)

Abstract— As the IoT ecosystem continues to grow, edge computing is becoming essential for handling and analyzing the vast amount of data generated by connected devices. Unlike traditional centralized data models, where information is sent to remote centers for processing, edge computing processes data closer to where it is generated. This decentralized approach helps reduce latency, optimizes bandwidth usage, and improves both privacy and security. However, the rise in IoT devices and the spread of edge computing also increase the potential for cyberattacks, demanding more robust security measures. With AI and machine learning being utilized to analyze IoT data, edge computing facilitates this analysis directly at the data source, pointing to a future where AI and ML applications are more prevalent on edge devices.

Keywords: *IoT, Edge computing, Cyberattacks, Artificial intelligence, Machine learning.*

1. INTRODUCTION

With the rapid development and acceptance of the Internet of Things (IoT), big data, and 5G network architecture, traditional cloud computing still needs to meet the ever-increasing data volume generated by network edge devices and the need for real-time services. The evolution of edge computing (EC) enables data processing near or at the network's edge, thus reducing the computational and communication overload. However, due to the exclusive benefits and characteristics of EC, such as heterogeneous distributed architecture, data processing, parallel computation, location awareness, and the need for mobility support, traditional data security and privacy mechanisms in cloud computing are not capable of the EC paradigm [1]. IoTs have upgraded conventional, passive devices into sensible ones, allowing them to transmit considerable volumes of relevant data over the internet. Data processing and analysis capabilities within an IoT framework enable these devices to function autonomously with minimal human intervention. Artificial intelligence (AI)--based algorithms are employed to analyze the substantial volumes of data generated within IoT networks, enabling the delivery of value-added public services [2]. IoT services are assembled on a foundation of miscellaneous technologies in hardware and software. These services leverage various network technologies and communication protocols, which include radio frequency identification (RFID), near-field communication (NFC), ZigBee, Bluetooth, electronic product code (EPC), low-energy wireless communication protocols, barcodes, long-term evolution (LTE) advanced, AI, and wireless sensor networks (WSNs) [3].

Projections indicate that the global IoT market will experience a compound annual growth rate (CAGR) of 10.53% from 2019 to 2025 [4]. Cisco estimates that in 2030, over 500 billion devices will be connected to the internet. In the present day, the impacts and applications of IoT are particularly notable in areas such as environmental sensing, healthcare monitoring systems, logistics supply chain management, real estate construction, energy management, drone-based applications, the manufacturing industry, and various other fields. Securing the IoT systems is crucial as they continue to grow and integrate further into our daily lives. Despite its numerous benefits, IoT also poses serious security concerns for enterprises and individual users. Any device that is connected to the internet could act as a doorway to an extensive network, including sensitive data.

DOI: https://doi.org/10.69955/ajoeee.2024.v4i2.69

Interconnected devices aggravate security concerns further by exposing more security flaws and vulnerabilities. In the absence of appropriate security and privacy measures, potential attacks and security threats may outweigh IoT benefits and applications.

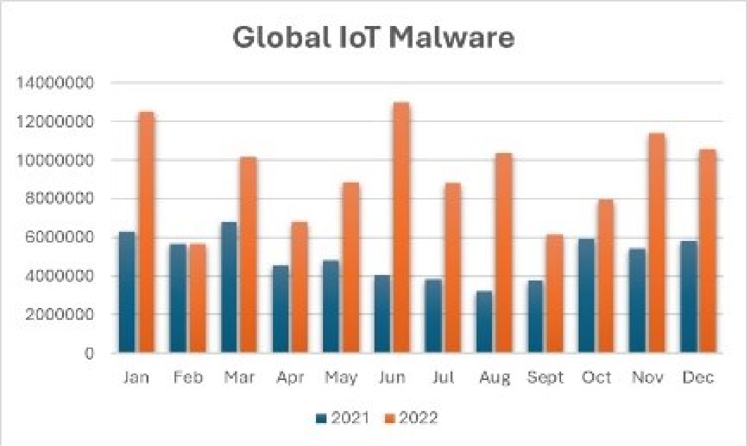


Fig. 1. Global IoT Malware during 2021 and 2022 [6]

Security solutions are expected to be lightweight that can be hosted on devices with lesser memory, computational abilities, and cost. Although numerous security solutions are proposed for standalone constrained devices, they are unsuitable for integration into the IoT network. The edge devices' heterogeneous nature, diverse computational capabilities, and network complexity necessitate lightweight security solutions that adhere to global standards [5]. The rapid expansion of IoT devices has opened new opportunities for cybercriminals. Security experts are frequently uncovering new malware targeting poorly secured IoT devices.

In 2022, SonicWall Capture Labs recorded 112.3 million instances of IoT malware, marking an 87% rise compared to 2021 (as shown in Fig. 1). Cyber attackers exploit IoT devices and networks to steal sensitive user data, including financial information, card details, location data, and health records. In edge computing-based (EC) IoT networks, significant amounts of user data are processed at the network's edge, spanning various industries and applications. The connection between edge devices and EC nodes is typically established through wired or wireless links. In contrast, EC nodes communicate with the cloud or data centers via public or private networks [7]. There are numerous cyberattacks targeting IoT applications. For example, the 2016 Mirai attack compromised over 2.5 million IoT devices and launched distributed denial of service (DDoS) attacks. Subsequent attacks, like Hajime and Reaper, further emphasized the security threats facing IoT devices [8]. As a result, developing security standards and guidelines for IoT is crucial to building secure and resilient IoT services. Regulatory bodies globally have also recognized the importance of IoT security [9].

This article provides a detailed review of IoT systems' security and privacy challenges, addressing associated technologies and protocols. It evaluates the current IoT architecture, identifies the security risks and limitations of underlying technologies, and concludes by summarizing key points on ongoing IoT security challenges, offering potential solutions.

2. EDGE COMPUTING

EC leverages present techniques, which ensures the processing of sensitive data at the network edge itself, thus managing the downstream data to centrally located cloud services as well as upstream data for IoT services. The "network edge" implies any computing or network resource between the data sources and the centrally located cloud-based data centers. The primary functions of EC include offloading computing jobs, data storage and caching, processing collected information, distributing user requests, and delivering cloud-based services closer to the end user. Although cloud computing has proven to be efficient for data processing due to its superior computational abilities power, the networks' bandwidth could not match the speed of data processing, forming a bottleneck for cloud-based computing. The concept of EC was conceived to place computing closer to data sources, offering several advantages over the traditional cloud-based computing approach. A comparison is presented in Table 1 [10].

Table 1: Comparing IoTs, Edge and Cloud computing [11]

	IoT	***Edge***	***Cloud***
Implementation	*Distributed*	*Distributed*	*Centralized*
Nature of the devices	*Physical*	*Edge nodes*	*Virtual nodes*
Computing capacity	*Less*	*Less*	*Larger*
Memory availability	*Very limited*	*Limited*	*Unlimited*
Response time	*N.A.*	*Fast*	*Slow*
Big data	*Source*	*Process*	*Process*

2.1 Edge Computing Architecture

The general architecture of EC is depicted in Fig. 2, representing edge computing MEC servers closer to the end users as compared to cloud-based data centers. Despite lower computational abilities, EC servers can offer better quality of service (QoS) and lower latency than cloud servers. The generic architecture of EC can be divided into three layers: the front-end, near-end, and far-end. The characteristics of each layer in an EC architecture are discussed below [11].

2.1.1. Front End

The Front-End layer consists of end devices such as sensors and actuators that manage data flow between two networks, functioning primarily as gateways for data entry or exit. Edge devices in this layer handle tasks such as data transmission, routing, processing, monitoring, filtering, translation, and storage as user data moves between networks. Edge computing (EC) capitalizes on the computing power of nearby end devices to provide real-time services for specific applications. However, since end devices have limited processing capacity, they often rely on server resources to meet most service requirements.

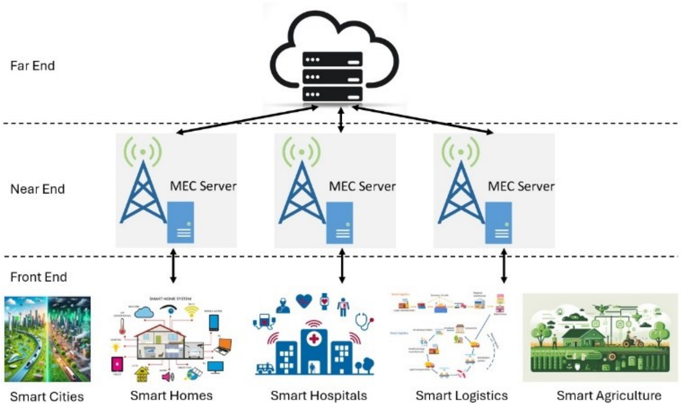


Fig. 2. Edge computing architecture.

2.1.2. Near End

The Multi-access Edge Computing (MEC) model and gateways in the near-end environment are designed to move technology resources closer to client devices and end users. Edge servers in this layer handle real-time data processing, data caching, and offloading computation tasks, offering computing and cloud-like services at the network edge. This reduces reliance on centralized cloud services for these processes.

2.1.3. Far End

The far end of the EC architecture consists of cloud data centers, including a centralized data hub and interconnected regional centers. These cloud data centers act as the ultimate repository for information. Since cloud servers are located far from the end devices, transmission latency becomes critical when delivering large-scale parallel data processing and storage services.

2.2 Edge computing benefits

According to the estimates published in Gartner report, about 75% of the network data produced at the business houses will be shifted out from the centrally located data centers for processing, a substantial increase from the 10% predicted in 2018. This trend demonstrates the growing adoption and acceptance of EC. By shifting computing resources and intelligence closer to the network's edge, EC offers numerous benefits, such as significantly lower latency, increased bandwidth, and enhanced privacy and security [12], [13]. These benefits of EC are further steering the adoption of various services such as IoT/M2M, 4K Ultra High Definition (UHD) video services, and mobile serious gaming. Also, MEC can offer application providers local context awareness, including Radio Access Network (RAN) analytics, traffic characteristics, and device location information [14]. Thus, EC solves latency-related challenges and supports users to optimize the benefits of cloud computing architectures. Forms of EC include local devices, localized data centers, and regional data centers. The benefits of EC can be summarized as,

Quicker data processing and analysis: EC minimizes the necessity for data transmission to centrally located cloud data centers, thus quicker response times and real-time processing. EC characteristics are leveraged in applications requiring rapid feedback, such as automatic driving, intelligent manufacturing, and video monitoring.

Security: EC processes the user data locally, mitigating the risk of data loss or leakage associated with data transmission to the cloud.

Lower energy consumption and bandwidth cost: EC minimizes the dependence on broad network bandwidth and energy consumption due to data processing locally.

2.3 Edge Computing Challenges

The edge-based servers provide distributed computing resources at a small-scale level; thus, EC-based IoT services are scalable and able to meet demands in large-scale applications like smart cities or autonomous driving. However, integrating EC with IoT poses unique challenges, and a seamless and efficient approach is needed to bridge the gap between these two technologies. The three important challenges in EC-based IoT systems are summarized below:

Heterogeneous IoT infrastructure: The edge devices/ sensors are deployed in diverse environments with unique purposes. Hence, various hardware devices and communication protocols are needed. Also, the deployment architecture of these devices in the EC environment varies with the application type. Thus, there is a need to explore a cooperation architecture involving hardware devices, communication protocols, and established industry standards to unify this diversity.

Coordination between communication and computing: Coordination between communication and computing is a bottleneck in the success of EC-driven IoT services. The limited power and computational capacity of edge devices and servers limit the amount of workload that can be transferred to the edge servers. Hence, an orchestration mechanism should be in place that allocates the workload between edge servers and IoT devices at optimal communication and computation costs.

Complicated security and privacy issues: Adversaries target IoT devices and edge servers to gain access to user data or disrupt the services. EC-based IoT systems’ heterogeneity and constrained computing capability are the foremost challenges in ensuring security and privacy. Appropriate countermeasures like robust authentication and encryption techniques, secured communication protocols, regular updates to underlying software and firmware to patch vulnerabilities, and adherence to strict access control policies should be adopted and implemented to address these challenges.

3. DATA SECURITY AND PRIVACY CHALLENGES

EC requires outsourcing end-user private data to external service providers, such as cloud or edge data centers, leading to data ownership and control loss. This separation can result in data loss, leakage, unauthorized access, compromised confidentiality and data integrity. EC leverages various technologies, including offloading, virtualization, and outsourcing, that bring the computational tasks closer to data sources. Users’ data privacy is an important driver for security, with the International Telecommunication Union's Telecommunication Standardization Sector (ITU-T) defining privacy as the right of individuals to manage the collection, processing, and storage of their personal data and control its access. The data privacy requirement is often ensured through mechanisms like cryptography, which restricts access to authorized parties and inhibits unauthorized disclosure [1].

Several factors contribute to the increased attack surface in the EC model. Two primary concerns are hardware limitations and software heterogeneity. Devices and servers at the edge layer typically have less computing and storage capacity than cloud servers, making implementing robust security measures like firewalls challenging and leaving them more vulnerable to attacks. Additionally, the lack of standardization in protocols and operating systems across diverse edge deployments further increases the risk of security breaches. Security threats in edge computing (EC) are continuously evolving, mainly due to the frequent mobility of user devices. These security challenges stem from design flaws, misconfigurations, and implementation errors. Xiao et al. have categorized most EC security threats into four main types, as shown in Fig. 3: Distributed Denial of Service (DDoS) attacks, side-channel attacks, malware injection attacks, and authentication and authorization attacks. Corresponding mitigation strategies for these threats are detailed in Table II [17].

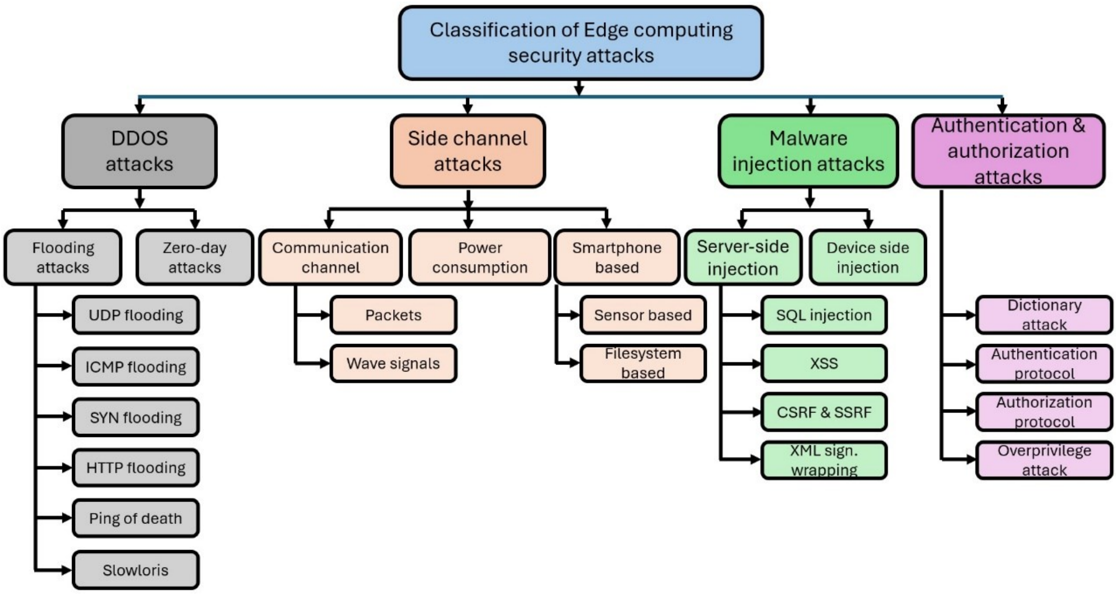


Fig. 3 Classification of EC security threats [15]

Distributed Denial of Service: DDoS attacks involve unauthorized server access through compromised edge devices. In these attacks, adversaries take control of edge devices and launch denial-of-service assaults on edge servers, effectively shutting down their services. Two common forms of DDoS attacks are zero-day attacks and flooding-based attacks. Flooding attacks overwhelm a server by bombarding it with many malicious network packets, such as UDP overflows, ICMP floods, SYN flash floods, HTTP flash floods, SYN flooding, ping of death, and delays, disrupting normal operations. Zero-day DDoS attacks are more advanced, relying on the attacker identifying an unknown vulnerability in the server's code. The attacker exploits this vulnerability, causing memory corruption and the eventual breakdown of the server's services. Flaws in communication network protocols primarily cause flooding attacks, while zero-day attacks exploit unaddressed vulnerabilities in server software [18].

Side-channel attacks exploit publicly accessible information about a target, known as side-channel data, rather than directly accessing sensitive information. Attackers use the correlations between the gathered public data and private information to infer the protected data. These attacks can occur at any point in the edge computing (EC) network, as public information can often be linked to sensitive data. For example, attackers may capture communication signals (such as packets or wave signals) to expose private user data or monitor the power consumption of edge devices to reveal usage patterns. Power analysis is a common technique for extracting side-channel data from EC networks. Power analysis-based attacks are categorized into two types: simple power analysis and differential power analysis. Simple power analysis involves closely examining individual power waveforms to extract valuable information. On the other hand, differential power analysis (DPA) consists of recording a series of power consumption readings while the device processes specific data, such as a secret encryption key. These readings are then compared to known power models to deduce parts of the secret key [19].

Table 2: Mitigation strategies against EC cybersecurity threats [15]

Station	*System Type*
DDoS attack	*Detect and filter technique is adopted. Individual packets are inspected to identify and remove the malicious content from the network. Machine learning and packet entropy can also help identify malicious packets. Countermeasures against zero-day attacks are difficult as the source codes are buried deep in the firmware.*
Side-channel attacks	*Data perturbation and differential privacy techniques. K-anonymity is the commonly used data perturbation technique that alters the identifier information before publishing sensitive attributes along with the data.*
Malware injection attacks	*The detection-and-filter technique has emerged as effective against server-side injection attacks. Defense strategies normally rely on static analysis to detect malicious code and implement a fine-grained access control mechanism.*
Authentication and authorization attacks	*Two common methods are improving the security of communication protocols and reinforcing cryptographic implementations to counter attacks on authentication protocols. To prevent over-privileged attacks, the most effective strategy is to enhance the permission models of operating systems on edge devices.*

A malware injection attack is a data security threat where attackers insert malicious code into a legitimate software application running on an edge server. This compromised software may lose functionality and potentially gain access to users' sensitive data. Such attacks exploit software vulnerabilities, allowing the attacker to run arbitrary code and take control of the targeted system for malicious purposes [20]. Due to the resource limitations of edge devices, they often lack robust firewalls, making them vulnerable to cybersecurity threats. Attackers can covertly install malicious software on an edge device or server. Server-side attacks are typically classified into four categories: SQL injection, cross-site scripting (XSS), XML signature wrapping, and Cross-Site Request Forgery (CSRF) or Server-Side Request Forgery (SSRF). Device-side attacks, on the other hand, commonly target the firmware of edge devices.

Authentication and Authorization attacks: Authentication is the process of confirming the identity of a user requesting access to services, while authorization defines the access rights and privileges of that user. In EC, authentication commonly occurs between edge devices and servers but can also happen between devices or servers in a decentralized system. Authorization involves the edge server granting access permissions to a particular device or its applications. Both processes in EC are susceptible to several types of attacks, which can be categorized into four main groups: dictionary attacks, attacks on authentication mechanism vulnerabilities, exploitation of authorization protocol flaws, and over-privileged attacks. [17]. Dictionary attacks use a list of access keys to bypass authentication systems. Authentication vulnerabilities are often exploited through weaknesses in security protocols like WPA/WPA2. Authorization attacks take advantage of poorly designed authorization protocols running in EC systems. In over-privileged attacks, attackers deceive the system to gain excessive access rights, allowing them to perform malicious actions within the EC network.

4. EDGE AI

Big data processing requires more powerful methods, such as AI technologies, to extract insights that enable better decisions and strategic business moves. Edge artificial intelligence, or edge AI, is the deployment of AI algorithms and models on edge devices like sensors or IoT devices. Edge AI facilitates real-time data processing and analysis without dependence on cloud computing infrastructure. Edge AI combines EC and AI technologies to execute machine learning (ML) algorithms on edge devices. Technologies such as self-driving cars, wearable devices, security cameras, and smart home appliances leverage edge AI capabilities to promptly provide users with real-time information. ML can control shared resources at the edge smartly and adaptively [21]. Edge AI offers several benefits. Firstly, it decentralizes the data required for refining algorithms. Secondly, it enables analysis and decision-making to be conducted close to the data source. From a security and privacy standpoint, edge AI can mitigate attack vectors by minimizing or eliminating data transfer between edge devices and their data centers. Training and execution of AI models on edge devices are confronted by several challenges and roadblocks discussed below [22].

Limited hardware capabilities: Edge devices are usually constrained by several factors, such as processing capability, data storage requirements, and network bandwidth, that limit the hosting of complex AI algorithms on edge devices.

Power constraints Mostly, edge devices support mobility, are battery-operated, and have low power, limiting their ability to perform intensive AI tasks.

Scalability issues Unlike cloud resources, the resources at the edge layer need to be more flexible to scale, and the heterogeneous nature of these resources can degrade service quality.

Collaboration challenges Coordination and cooperation between heterogeneous edge devices can be challenging, resulting in poor efficiency and effectiveness of AI models.

Data privacy concerns: Using original private data for model optimization on edge devices raises privacy concerns, and limited communication resources can restrict the distribution of computation to devices.

4.1 Hardware for Edge Devices

The algorithm and hardware selected for running a model on an edge device are crucial. Optimal hardware choice should consider accuracy, energy consumption, data throughput, and cost metrics. Edge devices designed for AI model execution can typically be categorized into four types based on their technical architecture [23].

Application-Specific Integrated Circuit (ASICs) Chip: ASICs are the best possible option for specific applications rather than general functions. Their smaller footprint, lesser power consumption, more robust security and performance make them ideal for meeting the demands of edge computing patterns for AI algorithms. On the other hand, Edge Tensor Processing Units (TPUs) are Google’s custom-designed chips used to accelerate Machine Learning workloads.

Graphics Processing Unit (GPUs): GPUs leverage the inherent data parallelism of mining programs to enhance throughput, achieving higher speeds compared to central processing units (CPUs). These GPUs' characteristics make them suitable for implementing AI algorithms, thus making them an ideal choice for designing and implementing edge devices. For example, NVIDIA’s Jetson TX1, TX2, and DRIVE PX2 are embedded AI computing devices equipped with GPUs. These devices offer a small form factor, lower latency, and low power requirements.

Field-Programmable Gate Array (FPGA): FPGAs are highly flexible, programmable hardware with lower energy requirements, parallel computing resources, and high security. Developers familiar with hardware description languages can quickly implement AI algorithms on FPGAs. However, FPGAs have poorer compatibility and more limited programming capabilities compared to GPUs. Leading FPGA manufacturers include AMD-owned Xilinx and Intel Altera.

Brain-Inspired Chip: Brain-inspired chips are constructed on a neuromorphic architecture, featuring programmable neurons on a silicon chip that process tasks akin to the human brain using synapses. These chips enable significantly accelerated processing of neural network applications in real-time, with extremely low power needs. Examples of neuromorphic processor chips include IBM True North and Intel Loihi, which are well-suited for complex AI algorithms.

5. CONCLUSION

Edge computing offers numerous benefits but also poses challenges that must be tackled. Security and privacy are foremost cause of concern as the user-sensitive data is processed and analyzed near edge devices. Implementing robust encryption, data protection, and secure communication protocols is crucial to mitigate these risks. Managing and scaling distributed edge infrastructure can also be complex, requiring seamless integration, network connectivity, and device management as edge device numbers increase. Standardization and interoperability across various edge computing solutions are essential for creating a cohesive and scalable ecosystem. Deploying machine learning on IoT devices reduces network congestion by enabling computations near data sources, ensuring data privacy, and lowering power consumption compared to continuous wireless transmission to central servers. The integration of specialized hardware into edge devices enhances computing efficiency in physical environments and improves responsiveness. Neuromorphic processors and sensors are also emerging, offering real-time intelligence and continuous onboard learning at the edge, even with a tight power budget, enabling complex AI computation at the network's edge.

REFERENCES

[1] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," IEEE Access, vol. 6, pp. 18209-18237, 2018. https://doi.org/10.1109/ACCESS.2018.2820162

[2] M. A. Albreem, A. M. Sheikh, M. H. Alsharif, M. Jusoh, and M. N. Mohd Yasin, "Green Internet of things (giot): Applications, practices, awareness, and challenges," IEEE Access, vol. 9, pp. 38833-38858, 2021. https://doi.org/10.1109/ACCESS.2021.3061697

[3] M. A. Albreem, A. M. Sheikh, M. J. Bashir, and A. A. El-Saleh, "Towards green internet of things (IoT) for a sustainable future in Gulf cooperation council countries: Current practices, challenges and future prospective," Wireless Networks, vol. 29, no. 2, pp. 539-567, 2023. https://doi.org/10.1007/s11276-022-03133-3

[4] M. A. M. Albreem, A. M. Sheikh, and A. A. El-Saleh, "Towards a sustainable environment with a green IoT: An overview," in 2022 International Conference on Computer Technologies (ICCTech), pp. 52-63, 2022. https://doi.org/10.1109/ICCTech55650.2022.00017

[5] P. M. Chanal and M. S. Kakkasageri, "Security and privacy in IoT: a survey," Wireless Personal Communications, vol. 115, no. 2, pp. 1667-1693, 2020. https://doi.org/10.1007/s11277-020-07649-9

[6] SonicWall, "2023 SonicWall cyber threat report." https://www.sonicwall.com/medialibrary/en/white-paper/2023-cyber-threat-report.pdf/, 2023. accessed:2024-04-26.

[7] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A survey on security and privacy issues in edge-computing assisted internet of things," IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4004-4022, 2020. https://doi.org/10.1109/JIOT.2020.3015432

[8] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: application areas, security threats, and solution architectures," IEEE Access, vol. 7, pp. 82721-82743, 2019. https://doi.org/10.1109/ACCESS.2019.2924045

[9] L. Jose, "Exploring IoT security issues and solutions." https://www.deviceauthority.com/blog/exploring-iot-security-issues-and-solutions/, 2023. accessed:2024-04-27.

[10] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, 2016. https://doi.org/10.1109/JIOT.2016.2579198

[11] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the Internet of things," IEEE Access, vol. 6, pp. 6900-6919, 2017. https://doi.org/10.1109/ACCESS.2017.2778504

[12] G. Singh, "Edge computing: Benefits and challenges." https://www.synopsys.com/blogs/chip-design/edge-computing-benefits-and-challenges.html, 2022. accessed:2024-04-27.

[13] A. Pradeep, "Exploring the future of edge computing: Advantages, limitations, and opportunities," in International Conference on Advanced Communication and Intelligent Systems, pp. 196-209, Springer, 2023. https://doi.org/10.1007/978-3-031-45124-9_15

[14] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration,"

IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1657-1681, 2017. https://doi.org/10.1109/COMST.2017.2705720

[15] M. S. Ansari, S. H. Alsamhi, Y. Qiao, Y. Ye, and B. Lee, "Security of distributed intelligence in edge computing: Threats and countermeasures," The Cloud-to-Thing Continuum: Opportunities and Challenges in Cloud, Fog and Edge Computing, pp. 95-122, 2020. https://doi.org/10.1007/978-3-030-41110-7_6

[16] S. A. Bhat, I. B. Sofi, and C.-Y. Chi, "Edge computing and its convergence with blockchain in 5g and beyond: Security, challenges, and opportunities," IEEE Access, vol. 8, pp. 205340-205373, 2020. https://doi.org/10.1109/ACCESS.2020.3037108

[17] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: State of the art and challenges," Proceedings of the IEEE, vol. 107, no. 8, pp. 1608-1631, 2019. https://doi.org/10.1109/JPROC.2019.2918437

[18] S. Nirenjena and D. Baskaran, "An investigation on distributed denial of service attack in edge computing," in 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 668-675, 2023. https://doi.org/10.1109/ICSSIT55814.2023.10061128

[19] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in Advances in Cryptology-CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15-19, 1999, Proceedings 19, pp. 388-397, Springer, 1999. https://doi.org/10.1007/3-540-48405-1_25

[20] K. Alsubhi, "A secured intrusion detection system for mobile edge computing," Applied Sciences, vol. 14, no. 4, p. 1432, 2024. https://doi.org/10.3390/app14041432

[21] T. Sipola, J. Alatalo, T. Kokkonen, and M. Rantonen, "Artificial intelligence in the iot era: A review of edge ai hardware and software," in 2022 31st Conference of Open Innovations Association (FRUCT), pp. 320-331, IEEE, 2022. https://doi.org/10.23919/FRUCT54823.2022.9770931

[22] C. Surianarayanan, J. J. Lawrence, P. R. Chelliah, E. Prakash, and C. Hewage, "A survey on optimization techniques for edge artificial intelligence (ai)," Sensors, vol. 23, no. 3, p. 1279, 2023. https://doi.org/10.3390/s23031279

[23] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, "A survey of recent advances in edge-computing-powered artificial intelligence of things," IEEE Internet of Things Journal, vol. 8, no. 18, pp. 13849-13875, 2021. https://doi.org/10.1109/JIOT.2021.3088875

ASIAN JOURNAL OF ELECTRICAL AND ELECTRONIC ENGINEERING
Vol. 4 No. 2 2024
E-ISSN: 2785-8189

A Systematic Review on Facial Detection and Recognition: Limitations and Opportunities

Mohammad Amir Khan*, Hasan Firdaus Bin Mohd Zaki, Azhar Bin Mohd Ibrahim,
S.M Afzal Hoq and Amir Akramin bin Shafie

Department of Mechatronics, International Islamic University of Malaysia, Kuala Lumpur, Malaysia

*Corresponding author: hasanzaki@iium.edu.my

(Received: 2 October 2024; Accepted: 4 October 2024)

Abstract— Face recognition technology is a biometric tool that identifies people by facial characteristics. Individuals collect the facial photography, which is then automatically processed by picture recognition software. Face detection and recognition have several potential applications in various departments like security, education, healthcare, etc. Therefore, the fundamentals and methods of broad facial detection and recognition have been discussed in this article. Owing to the outbreak of the pandemic, people are now required to wear masks so that the spreading of the coronavirus is prevented, which makes it challenging to monitor sizable crowds of mask-wearing individuals. Face masks have higher interclass similarities and interclass variability because they cover a significant portion of the face, fooling face recognition systems' facial verification process. Thus, this paper has also discussed various aspects of masked face recognition.

Keywords: *Detection, Recognition, Masked, Facial, Artificial Intelligence, Algorithms.*

1. INTRODUCTION

Face recognition is one of the most frequently used applications of picture analysis. A subset of the visual pattern recognition problem is face recognition. Humans constantly recognize visual patterns; our eyes are how we take in visual data. The brain interprets this information as meaningful concepts. Whether a picture or a movie, it is a matrix of many pixels for a computer. The computer should determine what idea every piece of data in data is referring to. A primary classification issue in visual model recognition exists here [1]. To do face recognition, it is crucial to identify the face's owner in the portion of the data that all machines understand as the face. It's a subdivision issue. Globally, the COVID-19 coronavirus disease has tremendously impacted people's daily lives. The World Health Organization has advised individuals to wear face masks in all public areas to stop the spread of COVID-19 [2]. It can be challenging to check people in public places for face masks manually. Furthermore, using face masks renders conventional face recognition methods, generally created for exposed faces, ineffectual. Thus, there is a pressing need to create a reliable system that can identify various people while donning a face mask and identify those who are not.

Although it is starting to be used in other sectors like logistics, retail, smartphone, transportation, education, real estate, government management, entertainment advertising, network information security, and others, face recognition is most frequently used in attendance, access control, security, and finance [3]. Face recognition can be used in security to identify offenders and alert authorities to potentially dangerous circumstances. Since artificial intelligence technology has advanced significantly, we now need recognition technology that is more precise, adaptable, and quick. Face detection is the first step in all facial analysis algorithms, including face alignment, face modeling, face relighting, face identification, face verification, authentication, head pose tracking, and recognition of facial emotions, gender, and age, among others [4].

This paper thoroughly analyzes face detection and the numerous face detection techniques in section 2. Masked face detection and various data sets used for face detection are also discussed in this section. In addition, challenges and applications of Face Detection have also been presented. After this, systems for generic facial recognition are examined in section 3. The development stages of facial recognition, the approaches taken for facial recognition, and the evaluation criteria of a facial recognition system are discussed in this section. Section 4 presents a detailed analysis of Masked Facial Recognition Systems. Recent advancements in Masked Facial

DOI: https://doi.org/10.69955/ajoeee.2024.v4i2.70

Recognition have been presented in section 5. The authors also provide a gap and some limitations of the current systems at the end of this section. Lastly, section 6 concludes the article.

2. FACE DETECTION

The initial step in face recognition is face detection. Face detection locates and measures a human face in a digital image. In the digital image, all additional objects like bodies, trees, and buildings are disregarded in favor of identifying facial characteristics. It might be considered a specific example of object-class identification, where the task is locating and figuring out the dimensions of every object in an image that fits into a particular class [5]. There are certain restrictions in advanced unconstrained situations, although general face detection algorithms can be applied to some extent. A vast area being obscured makes it challenging to detect occluded faces in unrestricted contexts because intraclass similarity and intraclass variation increase [6]. Utilizing adaptive technologies, numerous strategies are developed to tackle the issue successfully.

When faces are obscured, the current face detection models occasionally perform less accurately. They developed unique methods for obscured face detection to solve this issue. Due to intraclass similarity and intraclass variability, occluded object detection in unconstrained environments can be difficult when a significant portion of the environment is obscured. Numerous methods are used for obscured object detection. In a lab setting or interior situations, specialized methods like MTCNN, Sckit-Image, and Haar cascades can produce good results for detecting obscured faces. The convolutional correlational filter effectively addresses the issue—the same problem benefits from binary segmentation [7].

2.1 Masked Facial Detection Methods

We mainly concentrate on masked facial detection techniques in this area. Various Masked Facial Detection techniques are depicted in Figure 1. All methods are categorized using two kinds based on their characteristics: manually created feature-based methods and neural network-based methods. Standard methods also frequently include custom feature-based techniques. Depending on how many detectors are used, they can be further divided into two categories: single-detector and multiple-detector methods [8]. Most detectors use the AdaBoost algorithm [9]. Various detectors, such as the face detector, facial mask detector, nose and mouth detector, eye detector, and mouth and nose detector, are chosen or combined. Several researchers are interested in neural network-based techniques. The approaches can be divided into three groups based on how many stages they have: single-stage methods, two-stage methods, and multistage methods. They are primarily implemented for single-stage procedures through transfer learning of object detection algorithms. YOLO series methods, such as YOLO, YOLOv2, YOLOv3, YOLOv4, YOLOv5, and equivalent micro variants, are an example [10].

Regarding the utilization of neural networks, two-stage approaches can be further separated into three categories: neural network + neural network, neural network + hand-crafted feature, and hand-crafted feature + neural network. Face region classification and face region pre-detection are the two components of two-stage techniques. The first component is used to identify potential facial locations, while the second half is utilized to categorize the circumstances surrounding mask use. More sophisticated processing stages or employing several models in multistage approaches result in higher computation costs [11].

2.2 Data sets

Many datasets are proposed by academics worldwide to train detection or classification methods to observe the conditions of wearing masks. These models will be used in edge nodes or monitoring systems. Detailed descriptions and explanations of some of these datasets are provided in this section.

First, we will provide a previous dataset on masked face detection. 2017 Ge et al. [12] suggested the extensive MAFA dataset. It was asserted to be the largest mask-wearing dataset existing before 2017. Thirty-five thousand eight hundred six masked faces can be found among 30 811 online photographs that make up MAFA. The dataset is probable to have occluded faces since it includes a variety of masks, such as single-colored manufactured objects, hands, neckerchiefs, hair, and medical masks. Six attributes are assigned to the dataset: face orientation, eye position, mask location, face location, occlusion degree, and type of mask. It delivers appropriate samples and considers roughly 60 situations with masked faces. Many occlusions, meanwhile, are ineffective at shielding patients from infection threats.

The "MaskedFace-Net Image Dataset (MFNID)" was created by Cabani et al. [13] to produce simulated correct/incorrect masked faces. Candidate face detection, facial landmarks detection, mask-to-face mapping, and manual image filtering are the four phases that make up the system. Roy et al. [14] used photos from the Internet to compile the Moxa3K dataset. There are 3000 photos in total. The dataset considers boundary conditions carefully; for instance, if a handkerchief covers a face, it will be in the "mask" class. Many diverse samples are included in Moxa3K, including blurry, rotating, crowded, and lighting-related samples. All the facial areas are annotated in Pascal VOC format "LabelImg" and YOLO format, with 9161 faces and 2015 masked faces included. As a result, it gives researchers more options for how to train their machine-learning models. The robustness of masked facial detectors should increase in this circumstance.

By assembling photos from the publicly accessible datasets FFHQ [15], LFW [16], CelebA [17], YouTube videos, and the Internet, Eyiokur et al. [18] presented an Unconstrained Face Mask Dataset (UFMD). UFMD is a complicated collection that encompasses ethnicity, age, gender, and indoor and outdoor conditions, thanks to these publicly available photos. To increase the robustness of masked face detectors, UFMD also considers many head posture variations. The UFMD consists of 21 316 photographs divided into three classes: 10 618 images with faces that are masked, 10 698 images not masked, and 500 images with the wrong masks on. The website, according to the writers, will be accessible soon. Singh et al. manually created seven thousand five hundred images—5191 training photos, 1599 validation images, and 710 test images [19]. These pictures are from MAFA [12] and Wider Face [20]. The two classes "face" and "face-mask" in Singh's dataset are used to train a model to assess whether someone is wearing a mask. The extent of the crowing can be evaluated using the detection results. Annotations are supplied for bounding boxes. Table 1 provides a summary of these datasets.

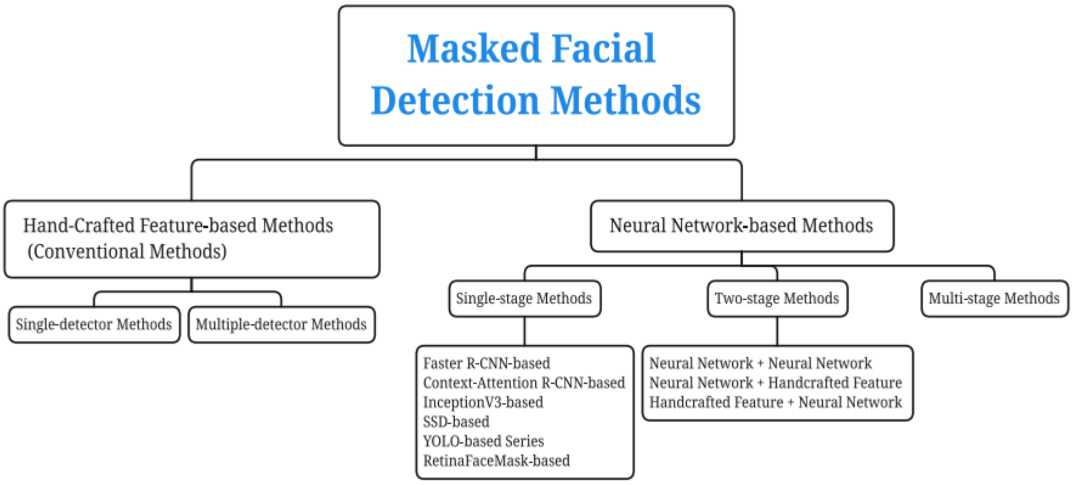

Masked Facial Detection Methods
Hand-Crafted Feature-based Methods (Conventional Methods)
Neural Network-based Methods
Single-detector Methods
Multiple-detector Methods
Single-stage Methods
Two-stage Methods
Multi-stage Methods
Faster R-CNN-based
Context-Attention R-CNN-based
InceptionV3-based
SSD-based
YOLO-based Series
RetinaFaceMask-based
Neural Network + Neural Network
Neural Network + Handcrafted Feature
Handcrafted Feature + Neural Network

Fig. 1. Masked Facial Detection Methods [11]

Table 1. Summary of the Datasets

Dataset name	Main characteristics	Image reality	Image number	Category	Masks number	Scale	Head pose	Scene	Annotation	Open
[12]	The pictures are all taken from the Internet. For each face region, six attributes are manually annotated.	Real	30811	Multiple mask types	35806 masked faces	Medium -large	Various	Complex	Yes	Yes
[13]	Face images are from FFHQ	Simulated	137016	Two	Sixty-seven thousand one hundred ninety-three faces with correct masks. 69823 faces with incorrect masks	Large	Frontal	Simple	No	Yes
[14]	The images are from the Kaggle	Real	3000	Two	9161 faces without masks.	Small Mediu	Various	Complex	Yes	Yes

Dataset name	Main characteristics	Image reality	Image number	Category	Masks number	Scale	Head pose	Scene	Annotation	Open
	data set, which has images captured in Russia, Italy, China, and India during a pandemic				3051 masked faces	m Large				
[18]	The pictures come from YouTube videos, FFHQ, LFW, CelebA, and other online sources.	Real	21316	Three	10698 faces without masks,10618 correct masked, 500 incorrect	Large	Frontal to Profile	Medium	Yes	Soon Open
[19]	The data set includes MAFA, WIDER, and FACE.	Real	7500	Two	5191 training images, 1599 validation images, 710 testing images	Small Medium Large	Various	Complex	Yes	Yes

2.3 *Challenges in Face Detection*

Face detection problems lead to a deterioration in face detection accuracy and rate of detection. Complex backgrounds, excessive faces in photographs, strange expressions, illuminations, low resolution, and face occlusion (hiding the face by any object) are a few of these difficulties. The variables considered are skin tone, altitude, proximity, hat, scarf, hand, hair, and any other thing, Background complexity, excessive faces, the image's orientation, etc. An example of this can be seen in Figure 2 [21].

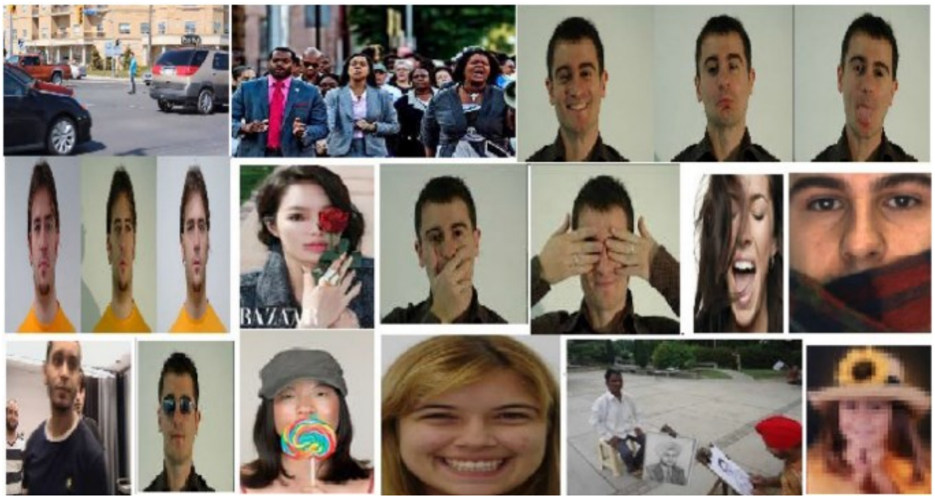


Figure 2. Challenges in face detection [21].

2.4 *Applications of Face Detection*

Gender classification, document and access control, biometric attendance, human-computer interface, face detection for autofocus in some modern digital cameras, photography, facial feature extraction, and marketing are some examples of applications for face detection. A webcam can be built into a television and detect any face that walks by. The approach next establishes the age range, gender, and race of the face. A sequence of advertisements specific to the indicated race, gender, and age can be played after collecting the data. [22].

3. GENERIC FACIAL RECOGNITION SYSTEMS

People started looking into ways to program machines to detect faces in the 1950s. Face geometry was primarily used for face recognition, as applied research in face recognition engineering began in 1964. The process below can create a face recognition system [23]. Identifying a specific area of a picture as a face is known as face localization or detection. This method has many applications, including face tracking, location estimation, and compression. Face normalization is one of the most important considerations when using a vector of geometrical features [24]. They must be somehow normalized to make the retrieved features independent of the face's position, scale, and rotation in the image plane. The next step in the feature extraction process is getting hold of pertinent facial characteristics from the data. The feature extraction procedure needs to be efficient regarding memory and processing time.

The feature extraction method includes dimension reduction, feature extraction, and feature selection. Any pattern recognition algorithm has to conduct dimensionality reduction [25]. The number of test images, features, and the complexity of the classifier all affect how well it performs. After feature extraction, feature selection is frequently carried out [26]. Features are first extracted from the face photos to identify the best subset of characteristics. Lastly, the face should be recognized by the system. The system would expose an identity from a database while performing an identification task. An accuracy metric, a classification algorithm, and a comparison method are used in this stage. Figure 3 shows the structure of the generic Face Recognition system.

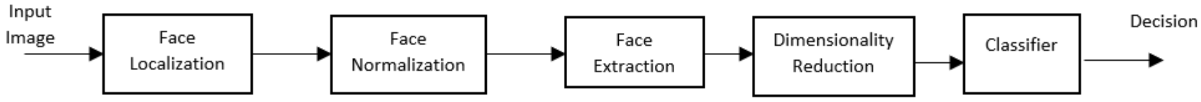


Figure 3. Generic Recognition

3.1 *Development stages of Facial recognition*

3.1.1 Early algorithm stage

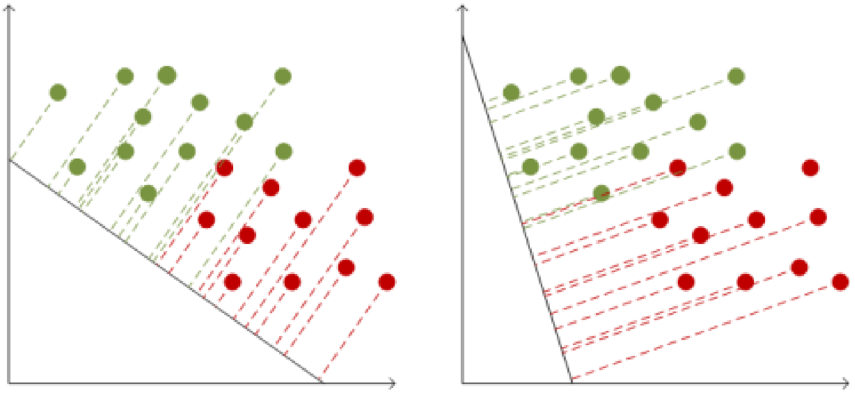


Figure 4. Comparison between PCA and LDA [29]

PCA- The principal component analysis (PCA) algorithm is the one most frequently used to reduce the dimensionality of data. PCA uses feature face extraction in algorithms for facial recognition. In 1991, MIT Media Laboratory's Turk and Pentland developed principal component analysis for facial recognition [27].

LDA - can employ linear discriminate analysis (LDA) for face recognition datasets with labels. It is applied in the classification process. As shown in Figure 4, While PCA requires that the data variance after dimensionality reduction be as significant as possible so that the data can be divided as widely as possible, LDA requires that the variance within the same category of data groups after projection be as small as possible and the variance between groups to be as large as possible [28].

3.1.2 Artificial features and classifier stage

SVM- Vapnik and Cortes introduced the support vector machine (SVM) in 1995. A method known as a support vector machine was explicitly created for high-dimensional, small-sample facial recognition problems. It is a classifier that was created using the generalized portrait algorithm. Due to its superior text classification performance, it quickly gained popularity as a machine learning technology [30]. AdaBoost- Schapire made the initial suggestion for boosting the algorithm. It is used to find faces. Any learning algorithm can be more accurate by using a boosting method. The fundamental concept is to combine various classifiers into a better final classifier by a few straightforward principles to improve overall performance [31].

SMALL SAMPLES - The term "small sample problem" alludes to the fact that there aren't enough face recognition training examples available, which prevents most face recognition algorithms from performing as well as they should. Numerous studies have been conducted to store picture information, maintain the association between samples, mitigate the effects of noise, and improve the face recognition effect more efficiently. To address the issue of limited sample sizes, Howland et al. devised a method that combined linear discriminant analysis with unspecified singular value decomposition (GSVD) [32].

NEURAL NETWORKS - An algorithm known as a neural network was created to replicate the human brain for face recognition. Face recognition has grown to be one of the research areas in the field of neural networks as one of the most crucial biometric identification techniques. Figure 5 depicts a typical neural network structure [33].

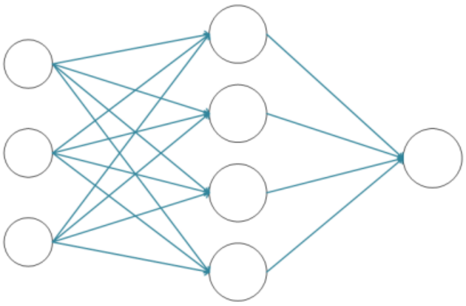
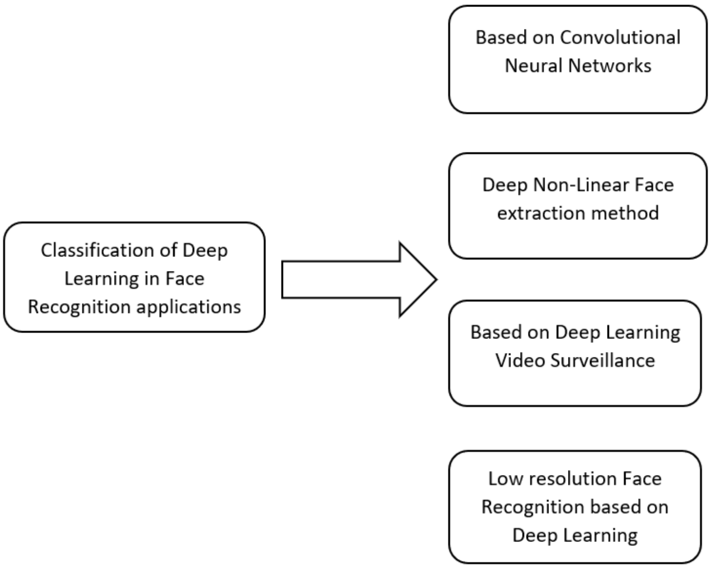


Figure 5. Typical Neural Network Structure [29]

3.1.3 Deep Learning

A subset of machine learning is deep learning. Without the requirement for feature extraction stages, deep learning may automatically identify the features required for categorization during the training phase. This is done to make network learning acquire more valuable features for face recognition. Deep learning has drastically changed the face recognition industry. Deep learning, frequently used in face recognition, can be broken down into the following categories, as shown in Figure 6. By integrating local perception areas, shared weights, and downsampling of face images, CNN optimizes the model structure using the data's locality and other variables. The complexity of the face's shape and texture contributes to the challenge of facial recognition. The authors in [34] presented a deep nonlinear face shape extraction approach from coarse to fine (coarse-to-fine auto-encoder networks, CFAN) further to enhance the nonlinear regression capacity of the algorithm to acquire hardiness to changes such as form.



Classification of Deep Learning in Face Recognition applications

Based on Convolutional Neural Networks

Deep Non-Linear Face extraction method

Based on Deep Learning Video Surveillance

Low resolution Face Recognition based on Deep Learning

Figure 6. Deep Learning in Face Recognition

For video search and surveillance, it's crucial to identify people reliably and rapidly in the footage. Schofield et al. devised a deep convolution neural network technique to examine animal behavior that could automatically recognize, track, and record human faces in the video [35]. Fourthly, deep learning-based low-resolution face recognition. The gathered face photos in practical applications frequently alter their position, and the limited image resolution causes a quick fall in face image identification performance. To enhance low-resolution face recognition, the most cutting-edge supervised discriminant learning approach was adopted [36], and the generative confrontation network pre-training method and the entire convolution structure were included.

3.2 Approaches for Facial Recognition

3.2.1 Feature Based approaches

Feature-based face recognition selects a few features to solely identify people using prior knowledge or local traits of faces. Local features chosen from facial photos include the eyes, nose, mouth, chin, and head contour. Topological graphs are used to express relationships between features [37]. A straightforward deterministic graph-matching algorithm takes advantage of the fundamental structure to extract recognizable faces from a database. Elastic Bunch Graph Matching is a technique that matches the gallery set, which serves as the model face graph, to the probe set, which serves as the input face graphs, to identify faces. The idea of nodes is fundamental to Elastic Bunch Graph Matching. A particular face feature point represents each node in the input face graph. For instance, one node might represent an eye, another might represent a nose, and so on to describe the other elements of the face. As shown in Figure 7, a graph-like data structure fitted to the face's shape is created by connecting the nodes for the input face graph [38].

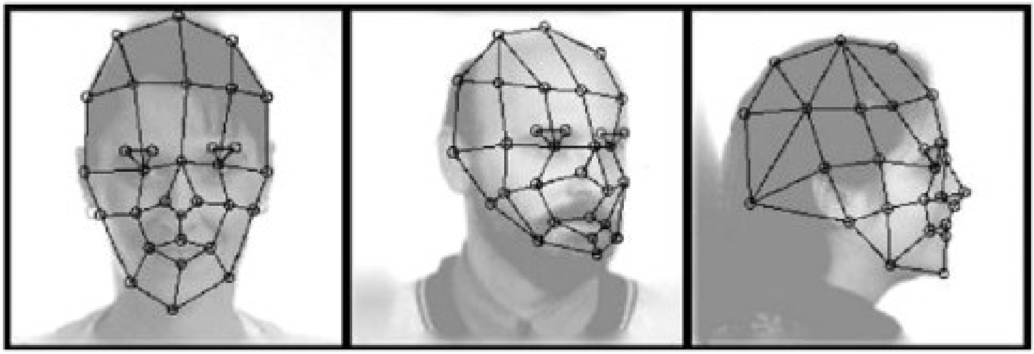


Figure 7. Face Topological graph [38]

3.2.2 Appearance Based approaches

Many academic fields, including biometrics, pattern recognition, and computer vision, have paid attention to appearance-based face identification approaches. Remarkably, two classifications—holistic and hybrid approaches—are implied. The entire face region is used as the initial input in the holistic approach to a recognition system. Additionally, it tries to capture the best possible representation of a face image as a whole and to use statistical patterns in pixel intensity variations [39]. Eigenimages, which come from principal component analysis, are one of the most often utilized representations of the facial region. The second group includes hybrid strategies; just as the human perception system uses local features and the entire face region to identify a face, a machine identification system should do the same.

3.2.3 Soft Computing based approaches

Soft computing tools can also be used for face recognition. Soft computing techniques like neural networks, fuzzy logic, and GA are commonly used. In face recognition, artificial neural networks are a standard tool [40]. They have been applied to categorization and pattern recognition. Because neural networks are nonlinear, their usage is enticing. Therefore, the feature extraction stage might be more efficient than linear KarhunenLoeve methods. The way a neural network is designed is crucial for effective recognition. It is highly dependent on the program being utilized. Higher recognition accuracy is provided by a face recognition system created using fuzzy logic and neural networks [41].

Artificial neural networks, fuzzy logic, and genetic algorithms (GAs) are examples of soft computing approaches that have become significant analytical methodologies in computer vision research. An artificial neural network is an effective technique for resolving the nonlinearity imposed by various restrictions. Fuzzy logic is also utilized to simulate human perception and thought. It is commonly known that analysis using fuzzy sets and fuzzy logic, as well as precise cognition, contribute to the efficacy of the human brain. Fundamental application limitations are almost always accompanied by uncertainty, a typical pattern recognition issue. Fuzzy logic-based analysis has demonstrated significantly improved results in pattern identification. GA is a potent optimization and search algorithm that is based on the notion of natural selection [42]. GA is effective at speeding up computation for a massive heap of space. Because recognizing faces from an ample heap space takes a lot of time, a GA-based strategy is employed to identify the unknown image quickly. GA is used when a user has insufficient time to provide findings without checking each database face. Faster face recognition will be possible with feature extraction and GA. Table 2 represents the summary of these approaches.

Table 2. The summary of these approaches

S.no	Approach	Description
1.	Feature-Based	selects a few features from faces based on past knowledge or local factors to identify people uniquely.
2.	Appearance Based	It is claimed that there are two categories: holistic and hybrid approaches.
3.	Soft Computing Based	We utilize soft computing methods like fuzzy logic, neural networks, and GA.

3.2.4 Algorithms for Face Recognition

Appearance-based or model-based algorithms can be used for facial recognition. The portrayal of the face in these techniques sets them apart. A face is represented using appearance-based approaches as a collection of raw-intensity images. Considered a high-dimensional vector, an image. A feature space is typically derived from the image distribution using statistical methods. This approach can be categorized as linear or nonlinear; linear appearance-based approaches reduce dimensions linearly. The complexity of nonlinear appearance algorithms is higher [1]. A nonlinear manifold is approximated by linear subspace analysis.

On the other hand, the model-based method [43] aims to model a human face. The calculated model's parameters identify the image when introducing the new sample. Model-based methods might be 2D or 3D in nature. These models are frequently morphable, allowing the classification of faces even in position alterations. Elastic Bunch Graph Matching or 3D Morphable Models are two examples of this methodology [44].

3.3 Evaluation Criteria for Facial Recognition

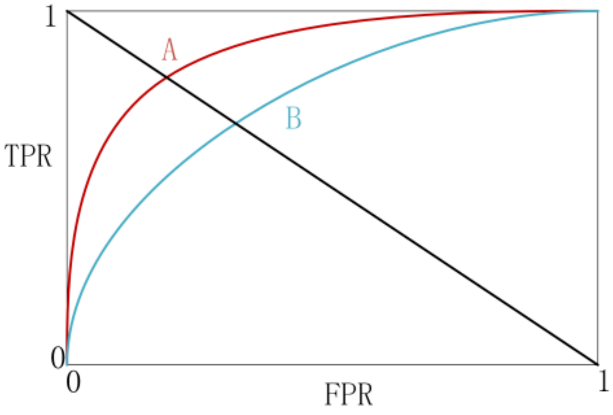


Figure 8. TPR FPR curve for two classifiers [29]

Three crucial metrics for assessing the effectiveness of the face recognition algorithm are accuracy (ACC), the ROC curve, and the area under the curve (AUC) value [45]. The ACC index is frequently used in facial recognition tasks. If M photographs are correctly identified out of the total N images in the testing set, The definition of ACC is as follows.

$$ACC = M/N \tag{1}$$

The algorithm's performance improves as the ACC value increases. In the face recognition task, the ROC first calculates the distance measurement or the similarity between photos. Then, it completes the recognition following the threshold to identify whether two photographs (also known as sample pairs) came from the same person [46]. The abscissa of the ROC curve represents the false positive rate (FPR), and the recall rate or valid positive rate is defined by the ordinate (TPR). The definitions of FPR and TPR are given as follows.

$$TPR = TP/(TP + FN) = FP/(FP + TN) \tag{2}$$

TP stands for the cheerful sample pair that the model incorrectly predicted, FN for the positive sample pair that the model incorrectly predicted, TN for the adverse sample pair that the model incorrectly predicted, and FP for the adverse sample pair that the model incorrectly predicted. Different TPR and FPR values can be obtained by adjusting different thresholds, and ROC curves can be produced (https://blog.csdn.net/). The red and blue curves in Figure 8 represent the TPR FPR curves of two distinct classifiers. The curve's point corresponding to a threshold value is depicted as the ROC curve. The method performs better when the ROC curve is near the upper left corner. In other words, it can still attain a high recall rate even with a meager incorrect recognition rate. AUC value, or the region below the ROC curve, is a scalar to assess the model's strengths. The higher the AUC value, the better the algorithm's performance (https://blog.csdn.net/) [29].

4. MASKED FACIAL RECOGNITION

The most challenging facial occlusion difficulty is recognizing masked faces since they obscure a significant portion of the frontal face, often 60%, which includes rich features like the nose and lips. Face masks deceive face recognition systems' facial verification processes by obscuring a significant portion of the face, resulting in higher inter-class similarities and interclass variance [47].

Matching a masked face with unmasked or masked faces is the goal of the masked face identification issue. When mask-wearing became crucial in containing the COVID-19 pandemic's transmission of the virus, this novel and complex study topic gained significant importance. MFR can support face recognition systems' ability to adapt to real-world situations, which is essential for maintaining public safety [48]. However, the researchers were forced to consider alternate solutions because the many mask types in this procedure made it impossible for the typical facial recognition systems to function correctly. Mask-wearing has become customary during this Covid-19 outbreak, and sadly, it appears that this will continue owing to new mutations. As a result, it is now difficult to deal with this issue efficiently and adequately [49]. The necessity for large-scale masked face datasets for training is revealed by deep learning being a promising contender in solving this complex problem. However, because these datasets are scarce and insufficient, researchers have begun using face-masking technologies to create artificially masked face datasets from existing large-scale face datasets. Masked face datasets can be made using two methods [50]. Collecting actual facial photos of people wearing masks is the first strategy. This strategy, however, takes a lot of effort and money. It is challenging to create an extensive image dataset. Finding the anchor points in the facial photos and adding artificial masks is a different strategy [51].

5. A RECENT INVESTIGATION IN THE FIELD OF MASKED FACIAL RECOGNITION

To address the critical biometric issue of masked face recognition—particularly in light of the continuing COVID-19 pandemic—the work in [52] presents a novel attention-guided deep CNN model. We suggest a dual-branch training method that incorporates our Upper Patch Attention (UPA) module as an auxiliary attention branch into the trunk CNN and directs the backbone network to improve the feature extraction capability in the non-occluded facial regions while maintaining the overall perception of a whole face image.

The Deepmasknet framework for precise face mask detection and masked facial recognition has been given in [53]. An internal unified MDMFR dataset has also been created to evaluate the effectiveness of the suggested strategy. To test face mask detection and masked facial recognition, we have assembled a sizable and varied collection of face photos. The accuracy of 93.33% for masked facial recognition and 100% for face mask detection have demonstrated our Deepmasknet model's superiority to existing methods. Additionally, test results on our MDMFR dataset and the three standard Kaggle datasets have confirmed the proposed Deepmasknet model's robustness for face mask detection and masked facial recognition under a variety of conditions, including variations in face angles, lightning conditions, gender, skin tone, age, and others.

A deep transfer learning model has been constructed [54] to recognize various forms of face masks. The suggested process has two key phasesData preprocessing (conversion of image format, ROI identification, and normalization) and data augmentation are included in the first step of the recognition process, which uses a deep transfer learning neural network (rotation, width, and height shifting, shearing, horizontal flipping, and pixel filling).

The work in [55] suggests and puts into practice a face mask detector that can determine whether a person's face is adequately protected by a mask in settings where COVID-19 regulations are in effect. This model used a two-stage detector construction. The model was trained via transmission learning with the MobileNetV2 framework on a dataset of 7514 photos. To extract the area of interest (ROI), a Haar cascade-based feature extractor and classification technique were used. This led to identifying the sections, which were then forwarded to the learning classification method to obtain the desired result. According to the results of the experiments, training accuracy was 98 percent, and testing accuracy was 97.45 percent.

The depth, azimuth, and elevation geometric properties of a 3D face are used in [56] to depict the face. The intrinsic benefits of 3D faces improve the 3D masked face recognition network's stability and viability. For the 3D masked face recognition network to fully utilize the depth, azimuth, and elevation information in differentiating face identities, a facial geometry extractor is also proposed to emphasize discriminative facial geometric traits. The DAE image stores depth, azimuth, and elevation data in three channels. This study suggests using FGE to draw attention to critical facial geometric elements on each channel. Figure 9 depicts the FGE's organizational structure. The DAE image's depth, azimuth, and elevation channel components correspond to a different convolution kernel. A single convolution kernel handles one channel, and one convolution kernel convolves a single channel. Each output feature map from one-to-one convolution emphasizes the associated geometric features on the face.

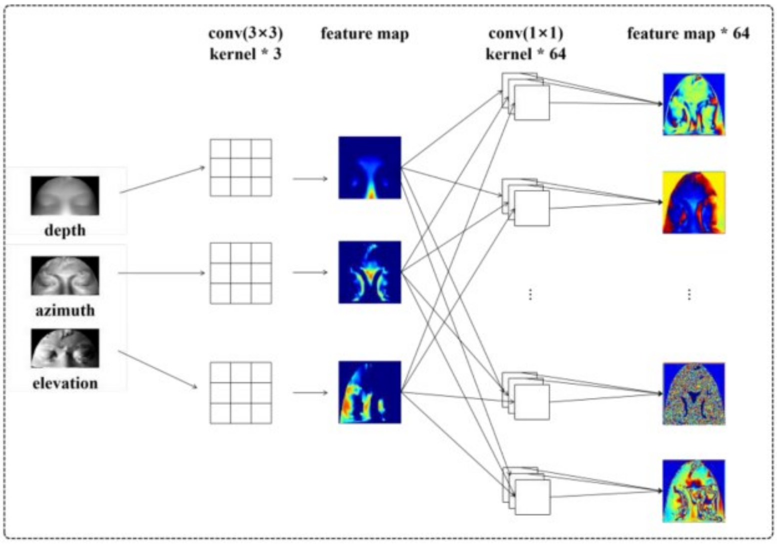


Figure 9. The structure of the Facial Geometry Extractor (FGE) [56]

In [57], the authors have put forth a method that enables us to keep using some facial recognition techniques for faces covered by safety masks, like those used for COVID-19. Our method comprises the preliminary processing of a picture with a hidden section. This preprocessing is based on determining the hue of the skin before processing. The recommended method entails first identifying the occluded and non-occluded facial regions and then distinguishing faces from skin regions. We validated and verified our method using FEI database face photos by introducing safety masks. We applied our strategy to enhance the functionality of the traditional Eigenfaces technology, which, when used directly, does not work on faces with protective masks. The research in [58] offers a straightforward method for achieving this goal using essential machine learning technologies, including TensorFlow, Keras, OpenCV, and Scikit-Learn. The proposed method correctly identifies the face in the picture or video and assesses whether it is wearing a mask. It can distinguish a face and a mask in motion and a video as a surveillance task performance. The method achieves superb accuracy. The authors investigate the best parameter settings for the Convolutional Neural Network model (CNN) to precisely detect the presence of masks without producing over-fitting.

The necessity for contactless biometrics during the COVID-19 pandemic has aided in the growth of masked face recognition. Due to the vast differences between entire faces and masked faces, it is challenging to obtain strong face representations when training a masked face recognition model. These two challenges must be addressed. The research in [56] suggests using non-masked face photos to train a 3D-masked face recognition network to address the first problem. This work uses the depth, azimuth, and elevation geometric properties of 3D faces to represent the face in the second problem. The intrinsic benefits of 3D faces improve the 3D masked face recognition network's stability and viability. A facial geometry extractor is also suggested to draw attention to distinguishing facial geometric aspects. The 3D masked face recognition network may fully utilize the depth, azimuth, and elevation information in differentiating face identities.

Table 4. Summary of recent work

Reference	**Methodology**	**Result**
[52]	A novel attention-guided deep CNN model is suggested.	The efficiency of the strategy has been extensively tested on synthetic and real-masked face datasets.
[53]	A Deepmasknet framework has been created for precise face mask detection and masked facial recognition.	The accuracy of 93.33% for masked facial recognition and 100% for face mask detection have demonstrated that our Deepmasknet model is better than existing methods.
[54]	A deep transfer learning model recognizes various types of face masks.	According to experimental data, the deep residual networks (ResNet101v2 and ResNet152v2) offer the best performance with the highest accuracy and negligible loss.
[55]	A face mask detector that can determine whether a person's face is covered adequately at locations where COVID-19 laws are enforced was created using a two-stage detector structure.	According to the results of the experiments, training accuracy was 98 percent, and testing accuracy was 97.45 percent.
[57]	The suggested method first identifies the occluded and non-occluded portions of the face and then distinguishes faces from skin regions.	The obtained experimental findings partially back the method the authors recommend for identifying face wear
[58]	These include TensorFlow, Keras, OpenCV, and Scikit-Learn, which are fundamental machine learning technologies.	The method can be applied to identify a face and mask in motion and video. The technique achieves superb accuracy.
[56]	The depth, azimuth, and elevation of the 3D face are used in this study's representation of the face's geometry. A facial geometry extractor is suggested to draw attention to distinguishing facial geometric traits.	The experimental results on four open 3D face datasets demonstrate that the proposed 3D masked face recognition network enhances the masked face recognition accuracy, proving that it is feasible to train the masked face recognition model using non-masked face photos.
[59]	The suggested approach is based on the FaceNet framework. It aims to improve the performance of both scenarios with and without a mask by altering the face recognition model currently in use.	The outcome displays a remarkable accuracy of 99.2% on a scenario with faces wearing masks.
[60]	A modified convolutional neural network creates a system for masked face recognition-based attendance (CNN).	The trial outcomes demonstrate the suggested CNN's high efficacy (i.e., 98.92%) in identifying masked faces for recording attendance.

In [59], a method for identifying human faces while wearing a mask is suggested. The lower third of the human face is obscured and cannot be utilized in the facial recognition learning process. Therefore, the proposed method is designed to recognize human faces on any facial features that are accessible, which may vary depending on whether a mask is worn or not—design, method, and strategy. The suggested approach is based on the FaceNet framework. It aims to improve the performance of both scenarios with and without a mask by altering the face recognition model currently in use. Then, simulated masked-face images are computed on top of the original face photos to aid in the learning process of face recognition.

Additionally, feature heat maps are created to show the bulk of the facial features that are important for identifying people wearing masks. In [60], a modified convolutional neural network creates a masked facial recognition-based attendance system (CNN). To accomplish this, CNN's original Softmax classifier is replaced with a Support Vector Machine. In a 5-fold cross-validation, the improved CNN's performance in identifying faces with masks was compared to that of other CNNs. The trial outcomes demonstrate the suggested CNN's high efficacy (i.e., 98.92%) in identifying masked faces for recording attendance.

Compared to cutting-edge automatic FR solutions, the research in [61] offers a joint evaluation and in-depth analysis of the face verification performance of human specialists. This incorporates four automatic recognition solutions and a thorough review by human experts. A series of key takeaways on various facets of the association between the verification behavior of humans and machines are presented at the study's conclusion. Table 4 provides a summary of the recent work.

5.1 Limitations and Gap

1. Mask-wearing conditions are ignored by some approaches, which merely identify the masked and non-mask facial classes.
2. It is common knowledge that inappropriate mask use will not stop the propagation of COVID-19. The detection of mask-wearing situations was only possible using a few methods. As a result, more algorithms should be tested to identify situations where people are putting masks over their faces.
3. There still needs to be standardized evaluation for many masked facial detection methods, even though some literature evaluates several methods. Several methodologies might be used on different platforms.
4. The outcomes presented in the original literature merely enable conceptual comparisons for readers. Making a fair judgment is difficult. Quite a few techniques can be used to produce a good performance.
5. Several approaches need more specific information regarding their cost-effectiveness and operating environment.
6. Running time is a significant measurement statistic in practical applications.
7. Using current ways to maintain performance with lightweight equipment is a challenge.

6. CONCLUSION

Face recognition technology has made enormous strides with the advancement of science and technology, but there is still potential for improvement in terms of practical use. The authors have reviewed a series of masked face detection methods. In addition to this, various challenges faced in facial detection have been highlighted. Apart from facial detection, this paper presents an overview of the recent work done in Facial Recognition systems. Considering the current pandemic outbreak, the authors have tried to pay special attention to masked facial detection and recognition. After carrying out a detailed review, it is safe to say that although great efforts are being taken in the field of facial detection and recognition or masked facial detection and recognition, detecting the proper mask-wearing conditions is still a concern.

REFERENCES

[1] T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, and T. Poggio, "A quantitative theory of immediate visual recognition," Prog. Brain Res., vol. 165, pp. 33–56, Jan. 2007. https://doi.org/10.1016/S0079-6123(06)65004-8

[2] N. Khan and S. Fahad, "Critical Review of the Present Situation of Corona Virus in China," SSRN Electron. J., Feb. 2020. https://doi.org/10.2139/SSRN.3543177

[3] P. Kaur, K. Krishan, S. K. Sharma, and T. Kanchan, "Facial-recognition algorithms: A literature review," https://doi.org/10.1177/0025802419893168, vol. 60, no. 2, pp. 131–139, Jan. 2020. https://doi.org/10.1177/0025802419893168

[4] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," Int. J. Comput. Vis., vol. 57, no. 2, pp. 137–154, May 2004. https://doi.org/10.1023/B: VISI.0000013087.49260.fb

[5] M. K. Hasan, M. S. Ahsan, Abdullah-Al-Mamun, S. H. S. Newaz, and G. M. Lee, "Human Face Detection Techniques: A Comprehensive Review and Future Research Directions," Electron. 2021, Vol. 10, Page 2354, vol. 10 no. 19, p. 2354, Sep. 2021. https://doi.org/10.3390/electronics10192354

[6] R. Ranjan et al., "A Fast and Accurate System for Face Detection, Identification, and Verification," IEEE Trans. Biometrics, Behav. Identity Sci., vol. 1, no. 2, pp. 82–96, Apr. 2019. https://doi.org/10.1109/TBIOM.2019.2908436.

[7] S. Hemathilaka and A. Aponso, "A Comprehensive Study on Occlusion Invariant Face Recognition under Face Mask Occlusion," Jan. 2022. https://doi.org/10.5121/csit.2021.111804

[8] S. Stein and G. A. Fink, "A new method for combined face detection and identification using interest point descriptors," 2011 IEEE Int. Conf. Autom. Face Gesture Recognition. Work. FG 2011, pp. 519–524, 2011. https://doi.org/ 10.1109/FG.2011.5771452

[9] H. Zeng, F. Qin, and K. Lin, "An Optimized Face Detection Based on Adaboost Algorithm," Proc. 2018 Int. Conf. Inf. Syst. Comput. Aided Educ. ICISCAE 2018, pp. 375–378, Mar. 2019. https://doi.org/ 10.1109/ICISCAE.2018.8666925

[10] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, "YOLO-face: a real-time face detector," Vis. Comput., vol. 37, no. 4, pp. 805–813, Apr. 2021. https://doi.org/10.1007/S00371-020-01831-7

[11] B. Wang, J. Zheng, and C. L. P. Chen, "A Survey on Masked Facial Detection Methods and Datasets for Fighting Against COVID-19," IEEE Trans. Artif. Intell., vol. 3, no. 3, pp. 323–343, 2022. https://doi.org/10.1109/TAI.2021.3139058

[12] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting masked faces in the wild with LLE-CNNs," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 426–434, 2017. https://doi.org/10.1109/CVPR.2017.53

[13] A. Cabani, K. Hammoudi, H. Benhabiles, and M. Melkemi, "MaskedFace-Net – A dataset of correctly/incorrectly masked face images in the context of COVID-19," Smart Heal., vol. 19, p. 100144, Mar. 2021. https://doi.org/10.1016/J.SMHL.2020.100144

[14] B. Roy, S. Nandy, · Debojit Ghosh, · Debarghya Dutta, · Pritam Biswas, and T. Das, "MOXA: A Deep Learning Based Unmanned Approach for Real-Time Monitoring of People Wearing Medical Masks," Trans. Indian Natl. Acad. Eng. 2020 53, vol. 5, no. 3, pp. 509–518, Jul. 2020. https://doi.org/10.1007/S41403-020-00157-z

[15] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks." pp. 4401–4410, 2019. https://doi.org/10.1109/CVPR.2019.00453

[16] G. B. Huang and E. Learned-Miller, "Labeled Faces in the Wild: Updates and New Reporting Procedures," Accessed: Dec. 21, 2022. [Online]. Available: http://vis-www.cs.umass.edu/.

[17] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild." pp. 3730–3738, 2015, Accessed: Dec. 21, 2022. [Online]. Available: http://personal.ie.cuhk.edu.hk/.

[18] F. I. Eyiokur, H. K. Ekenel, and A. Waibel, "Unconstrained face mask and face-hand interaction datasets: building a computer vision system to help prevent the transmission of COVID-19," Signal, Image Video Process., pp. 1–8, Jul. 2022. https://doi.org/10.1007/s11760-022-02308-x

[19] S. Singh, U. Ahuja, M. Kumar, K. Kumar, and M. Sachdeva, "Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment," Multimed. Tools Appl., vol. 80, no. 13, pp. 19753–19768, May 2021. https://doi.org/10.1007/s11042-021-10711-8

[20] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "WIDER FACE: A Face Detection Benchmark." pp. 5525–5533, 2016. https://doi.org/10.1109/CVPR.2016.596

[21] A. Kumar, A. Kaur, and M. Kumar, "Face detection techniques: a review," Artif. Intell. Rev., vol. 52, no. 2, pp. 927–948, 2019. https://doi.org/10.1007/s10462-018-9650-2

[22] D. S. Brar, A. Kumar, Pallavi, U. Mittal, and P. Rana, "Face Detection for Real World Application," Proc. 2021 2nd Int. Conf. Intell. Eng. Manag. ICIEM 2021, pp. 239–242, Apr. 2021, https://doi.org/10.1109/ICIEM51511.2021.9445287

[23] A. Review, "Past, Present, and Future of Face Recognition: A Review," 2020.

[24] K. Nagano et al., "Deep face normalization," ACM Trans. Graph., vol. 38, no. 6, Nov. 2019. https://doi.org/10.1145/3355089.3356568

[25] S. Ayesha, M. K. Hanif, and R. Talib, "Overview and comparative study of dimensionality reduction techniques for high dimensional data," Inf. Fusion, vol. 59, pp. 44–58, Jul. 2020. https://doi.org/10.1016/J.INFFUS.2020.01.005

[26] B. Venkatesh and J. Anuradha, "A review of Feature Selection and its methods," Cybern. Inf. Technol., vol. 19, no. 1, pp. 3–26, 2019, https://doi.org/10.2478/CAIT-2019-0001

[27] A. Alahmadi, M. Hussain, H. A. Aboalsamh, and M. Zuair, "PCAPooL: unsupervised feature learning for face recognition using PCA, LBP, and pyramid pooling," Pattern Anal. Appl., vol. 23, no. 2, pp. 673–682, May 2020. https://doi.org/10.1007/s10044-019-00818-y

[28] A. Ouyang, Y. Liu, S. Pei, X. Peng, M. He, and Q. Wang, "A hybrid improved kernel LDA and PNN algorithm for efficient face recognition," Neurocomputing, vol. 393, pp. 214–222, Jun. 2020. https://doi.org/10.1016/j.neucom.2019.01.117

[29] L. Li, X. Mu, S. Li, and H. Peng, "A Review of Face Recognition Technology," IEEE Access, vol. 8, pp. 139110–139120, 2020. https://doi.org/10.1109/ACCESS.2020.3011028

[30] P. VenkateswarLal, G. R. Nitta, and A. Prasad, "Ensemble of texture and shape descriptors using support vector machine classification for face recognition," J. Ambient Intell. Humaniz. Comput., pp. 1–8, Apr. 2019. https://doi.org/10.1007/S12652-019-01192-7

[31] K. Sowmya, "Facial Recognition for Automated Attendance System Using Ada Boost Algorithm."

[32] F. Xu, J. Gao, and X. Pan, "Cow Face Recognition for a Small Sample Based on Siamese DB Capsule Network," IEEE Access, vol. 10, pp. 63189–63198, 2022. https://doi.org/10.1109/ACCESS.2022.3182806

[33] S. Almabdy and L. Elrefaei, "Deep Convolutional Neural Network-Based Approaches for Face Recognition," Appl. Sci. 2019, Vol. 9, Page 4397, vol. 9, no. 20, p. 4397, Oct. 2019. https://doi.org/ 10.3390/APP9204397

[34] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-Fine Auto-encoder Networks (CFAN) for real-time face alignment," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8690 LNCS, no. PART 2, pp. 1–16, 2014. https://doi.org/10.1007/978-3-319-10605-2_1

[35] D. Schofield et al., "Chimpanzee face recognition from videos in the wild using deep learning," Sci. Adv., vol. 5, no. 9, Sep. 2019. https://doi.org/10.1126/sciadv.aaw0736

[36] P. Li, L. Prieto, D. Mery, and P. J. Flynn, "On Low-Resolution Face Recognition in the Wild: Comparisons and New Techniques," IEEE Trans. Inf. Forensics Secur., vol. 14, no. 8, pp. 2000–2012, Aug. 2019. https://doi.org/10.1109/TIFS.2018.2890812

[37] S. H. Oh, G. W. Kim, and K. S. Lim, "Compact deep learned feature-based face recognition for Visual Internet of Things," J. Supercomput., vol. 74, no. 12, pp. 6729–6741, Dec. 2018. https://doi.org/10.1007/S11227-017-2198-0

[38] L. C. Jain, U. Halici, I. Hayashi, S. B. Lee, and S. Tsutsui, "Intelligent Biometric Techniques in Fingerprint and Face Recognition," Intell. Biometric Tech. Fingerpr. Face Recognition., no. July 1997, pp. 1–463, 2022, https://doi.org/10.1201/9780203750520

[39] P. Payal and M. M. Goyani, "A comprehensive study on face recognition: methods and challenges," https://doi.org/10.1080/13682199.2020.1738741, vol. 68, no. 2, pp. 114–127, Feb. 2020. https://doi.org/10.1080/13682199.2020.1738741

[40] K. Amit, "Artificial Intelligence and Soft Computing : Behavioral and Cognitive Modeling of the Human Brain," Artif. Intell. Soft Comput., Oct. 2018. https://doi.org/10.1201/9781315219738

[41] Y. I. Daradkeh, I. Tvoroshenko, V. Gorokhovatskyi, L. A. Latiff, and N. Ahmad, "Development of effective methods for structural image recognition using the principles of data granulation and apparatus of fuzzy logic," IEEE Access, vol. 9, pp. 13417–13428, 2021. https://doi.org/10.1109/ACCESS.2021.3051625

[42] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on the genetic algorithm: past, present, and future," Multimed. Tools Appl., vol. 80, no. 5, pp. 8091–8126, Feb. 2021. https://doi.org/10.1007/s11042-020-10139-6

[43] J. Li, T. Qiu, C. Wen, K. Xie, and F. Q. Wen, "Robust Face Recognition Using the Deep C2D-CNN Model Based on Decision-Level Fusion," Sensors 2018, Vol. 18, Page 2080, vol. 18, no. 7, p. 2080, Jun. 2018. https://doi.org/10.3390/S18072080

[44] M. P. Beham and S. M. M. Roomi, "A review of face recognition methods," Int. J. Pattern Recognit. Artif. Intell., vol. 27, no. 4, pp. 1–35, 2013. https://doi.org/10.1142/S0218001413560053

[45] J. J. Howard, L. R. Rabbitt, and Y. B. Sirotin, "Human-algorithm teaming in face recognition: How algorithm outcomes cognitively bias human decision-making," PLoS One, vol. 15, no. 8, p. e0237855, Aug. 2020. https://doi.org/10.1371/journal.pone.0237855

[46] P. Grother, M. Ngan, K. Hanaoka, W. Ross, and W. Copan, "Face Recognition Vendor Test (FRVT) Part 3: Demographic Effects," 2019. https://doi.org/10.6028/NIST.IR.8280

[47] M. Alexiou, I. P. Ktistakis, and G. Goodman, "Towards a Masked Face Recognition Algorithm: A Novel Rule Based Hybrid Algorithm," 6th South-East Eur. Des. Autom. Comput. Eng. Comput. Networks Soc. Media Conf. SEEDA-CECNSM 2021, 2021. https://doi.org/10.1109/SEEDA-CECNSM53056.2021.9566244

[48] Y. Martínez-Díaz, H. Méndez-Vázquez, L. S. Luevano, M. Nicolás-Díaz, L. Chang, and M. González-Mendoza, “Towards Accurate and Lightweight Masked Face Recognition: An Experimental Evaluation,” IEEE Access, vol. 10, pp. 7341–7353, 2022. https://doi.org/10.1109/ACCESS.2021.3135255

[49] H. Du, H. Shi, Y. Liu, D. Zeng, and T. Mei, "Towards NIR-VIS Masked Face Recognition," IEEE Signal Process. Lett., vol. 28, pp. 768–772, 2021. https://doi.org/10.1109/LSP.2021.3071663

[50] K. Wang et al., "Mask Aware Network for Masked Face Recognition in the Wild," Proc. IEEE Int. Conf. Comput. Vis., vol. 2021-Octob, pp. 1456–1461, 2021. https://doi.org/10.1109/ICCVW54120.2021.00168

[51] J. Yu, X. Hao, Z. Cui, P. He, and T. Liu, "Boosting Fairness for Masked Face Recognition," Proc. IEEE Int. Conf. Comput. Vis., vol. 2021-Octob, pp. 1531–1540, 2021. https://doi.org/10.1109/ICCVW54120.2021.00178

[52] Y. Zhang, X. Wang, M. S. Shakeel, H. Wan, and W. Kang, "Learning upper patch attention using dual-branch training strategy for masked face recognition," Pattern Recognit., vol. 126, 2022. https://doi.org/10.1016/j.patcog.2022.108522

[53] N. Ullah, A. Javed, M. Ali Ghazanfar, A. Alsufyani, and S. Bourouis, "A novel DeepMaskNet model for face mask detection and masked facial recognition," J. King Saud Univ. - Comput. Inf. Sci., Volume 34, Issue 10, Part B, 2022. https://doi.org/10.1016/j.jksuci.2021.12.017

[54] R. Mar-Cupido, V. García, G. Rivera, and J. S. Sánchez, "Deep transfer learning for the recognition of types of face masks as a core measure to prevent the transmission of COVID-19," Appl. Soft Comput., vol. 125, p. 109207, 2022. https://doi.org/10.1016/j.asoc.2022.109207

[55] M. Gupta, G. Chaudhary, D. Bansal, and S. Pandey, "DTLMV2—A real-time deep transfer learning mask classifier for overcrowded spaces," Appl. Soft Comput., vol. 127, no. June 2020, p. 109313, 2022. https://doi.org/10.1016/j.asoc.2022.109313

[56] Y. Wang, Z. Yang, Z. Zhang, H. Zang, Q. Zhu, and S. Zhan, "Masked Face Recognition with 3D Facial Geometric Attributes," ACM Int. Conf. Proceeding Ser., pp. 13–21, 2022. https://doi.org/10.1145/3529446.3529449

[57] F. Ennaama, K. Benhida, and S. Essalki, "Proposed approach to improve facial recognition techniques for occluded faces by Covid-19 mask protection," IFAC-PapersOnLine, vol. 55, no. 12, pp. 456–461, 2022. https://doi.org/10.1016/j.ifacol.2022.07.354

[58] G. Kaur et al., "Face mask recognition system using CNN model," Neurosci. Informatics, vol. 2, no. 3, p. 100035, 2022. https://doi.org/10.1016/j.neuri.2021.100035

[59] W. Moungsouy, T. Tawanbunjerd, N. Liamsomboon, and W. Kusakunniran, "Face recognition under mask-wearing based on residual inception networks," Appl. Comput. Informatics, 2022. https://doi.org/10.1108/ACI-09-2021-0256

[60] Y. Bong and G. Lee, "International Journal on Robotics, Automation and Sciences," vol. 3, pp. 33–41, 2021. https://doi.org/10.33093/ijoras.2021.3.6

[61] N. Damer, F. Boutros, M. Süßmilch, M. Fang, F. Kirchbuchner, and A. Kuijper, "Masked face recognition: Human versus machine," IET Biometrics, vol. 11, no. 5, pp. 512–528, 2022. https://doi.org/10.1049/bme2.12077