JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A CNN-BASED FRAMEWORK FOR COMPARISON OF CONTACTLESS TO CONTACT-BASED FINGERPRINTS

Dr. V. Sulochana, P.Suresh, R D.Madana Gopalan

Professor, Student, Student PG & Research, Department of Computer Applications, Hindusthan College Of Arts And Science Coimbatore, India

ABSTRACT

Numerous methods are employed for fingerprint recognition, and each method is predicated on particular standards. Finding a reliable method for fingerprint recognition is the goal of this endeavor. The goal of this project is to provide a straightforward, high-performing method for fingerprint identification. This method is divided into two main phases: the actual data collecting of human fingerprint samples occurs in the first stage, and the design and implementation of a high performance fingerprint recognition method takes up the second. The applied method focused on the feature extraction phase, where high performance features are produced by applying multiple levels of the two-dimensional discrete cosine transform (2D-DCT). This strategy is carried out by combining the thumbs of the left and right fingers. The outcomes show that this method achieves a good level of recognition accuracy.

Keywords: OS Fingerprinting, Network Monitoring, Machine Learning

1.INTRODUCTION

1.1 OS FINGERPRINTING

Operating System (OS) fingerprinting is a crucial component that plays a crucial part in understanding and protecting a network in the vast field of computer networks and cybersecurity. Network administrators, security professionals, and bad actors use OS fingerprinting as a basic technique to find and gather information about the operating systems that are running on distant devices connected to a network. Operating systems define the capabilities, limitations, and usefulness of all computing devices. They are the foundation of all hardware. Understanding the operating system of a particular device is essential for several reasons, such as making sure compatibility, putting in place suitable security measures, and spotting possible security flaws.

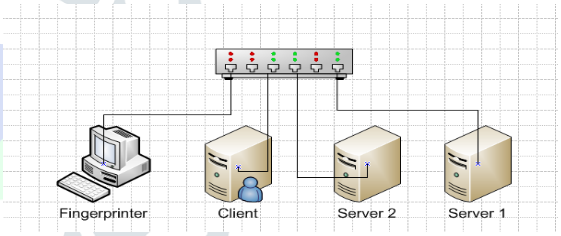

Figure 1 OS FINGERPRINTING

1.2 NETWORK MONITORING

Computer networks provide the essential framework for almost all aspects of our personal and professional life in today's globally connected society. Networks are essential for enabling communication, data interchange, and the smooth flow of information. Examples of these include the global web, which links people all over the world, and local area networks (LANs), which provide corporations and institutions more power. However, maintaining these networks' optimal performance, security, and dependability is a difficult task. This is the point at which network monitoring becomes essential. In the fields of network administration and information technology, network monitoring is a crucial procedure. It comprises the ongoing monitoring, evaluation, and control of a network's functionality, traffic, and components. Maintaining the network's functionality and health, proactively preventing and resolving problems, and eventually improving user experience are the main goals of network monitoring.

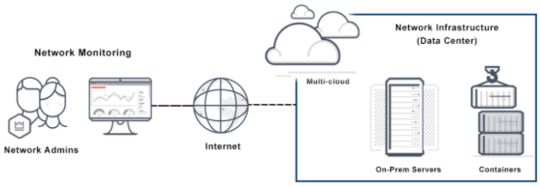

Figure 2 NETWORK MONITORING

1.3 MACHINE LEARNING

Within the discipline of artificial intelligence (AI), machine learning focuses on creating models and algorithms that allow computers to learn from data and make predictions or judgments. Its extensive use in a wide range of sectors, such as healthcare, banking, entertainment, and transportation, has recently led to its enormous popularity and significance. The purpose of this introduction is to give readers a basic grasp of machine learning, along with an overview of its major ideas and practical applications. A branch of artificial intelligence known as machine learning gives computers the capacity to learn from data and improve their performance on particular tasks without the need for explicit programming. In contrast to traditional programming, which involves giving a computer specific instructions to do a task, machine learning is teaching a computer patterns and rules from data in order to forecast or make judgments.

2. LITERATURE REVIEW

In their study, Yongxin Liu et al.[1] made the argument that the Internet of Things (IoT) is starting to play a major role in daily life by enabling the establishment of a wide range of services and applications. On the other hand, the existence of unapproved IoT devices has made the IoT vulnerable to serious hazards and tragic outcomes. Finding and classifying illegal IoT devices is the first step in securing the Internet of Things. Conventional methods make use of cryptographic techniques to confirm and validate the identities of trustworthy devices. Nevertheless, not all systems support these cryptographic protocols. Furthermore, these techniques are less successful when encryption keys are leaked or when devices that are legitimate can be exploited. As a result, non-cryptographic techniques for detecting rogue devices and identifying IoT devices have become effective ways to improve system security and give systems that use cryptographic protocols more defense. Non-cryptographic methods, however, need additional work and have not yet received enough attention.

The exponential rise of networked devices in the current era [2] of network-based computing has increased the diversity, faults, and cybersecurity dangers associated with these gadgets. The functionality and performance of emerging environments, like Industry 4.0, Smart Cities, and crowd sensing, depend on determining the capabilities of their devices, like sensors and actuators, and spotting potential misbehavior that might result from system errors, cyberattacks, or incorrect configurations. In order to do this, a potential area of research has arisen that focuses on modeling the behavior of the device's components as well as its operations through the creation and management of fingerprints. The current study, written by Pedro Miguel Sanchez et al., looks at the application scenarios, behavioral sources, processing, and evaluation methods that have contributed to the recent expansion of the device behavior fingerprinting area. In order to address the two main situations of device identification and device misbehavior detection, the study starts with a thorough analysis of the device kinds, behavioral data, processing, and assessment methods employed by the most recent and representative research works.

The system that Sunghyun Yuet al. proposed in [3]. emphasizes how crucial privacy protection is to computer communication. It draws attention to how easily plaintext communications without encryption can be intercepted and eavesdropped upon. As a result, the number of cyberattacks that take use of encrypted communication protocols has increased along with their adoption. Although decryption is essential to stopping these attacks, it comes with additional expenses and privacy hazards. The increasing use of network configurations that are independent of existing IP address schemes and the hazy boundaries of software-defined and cloud-based networks are expected to make existing network fingerprinting techniques—which rely on data from the TCP/IP stack—less effective. Our goal in this research is to examine and evaluate the Transport Layer Security (TLS) fingerprinting method, which enables encrypted communication to be classified and analyzed without needing to be decrypted. The shortcomings of the current network fingerprinting methods are addressed by this method. We include an overview of each TLS fingerprinting method as well as analysis data.

Hailey Selassie Hague al. has [4] put forth a plan for controlling and safeguarding extensive, intricate business network infrastructure. Real-time network traffic trace capture and analysis are part of this technology. Reliable operating system (OS) identification using passive fingerprinting is essential for efficient network administration and cybersecurity defense. Passive fingerprinting offers an advantage over active fingerprinting in that it lowers the possibility of false alarms because it doesn't add to the network load. This research proposes and evaluates an enhanced classification strategy to passive OS fingerprinting using the latest deep learning and classical machine learning techniques. Two distinct methodologies are used to conduct controlled studies on both mimicked and reality traffic in addition to benchmark data. It is determined that the underlying TCP variation is an important factor for predicting the remote OS through an Oracle-based machine learning approach. On the basis of this finding, an advanced OS fingerprinting tool is created.

In the framework of the Internet of Things (IoT), Destem Haile Selassie Hagset al.[5] has presented a solution that answers the demand for securing and controlling massive, complicated enterprise network infrastructure. The real-time acquisition and analysis of network traffic traces is the main function of this technology. Network management and cybersecurity prevention depend on a precise passive Operating System (OS) fingerprinting method. Passive fingerprinting offers an advantage over active fingerprinting in that it lowers the possibility of false alarms because it doesn't add to the network load. This work proposes a state-of-the-art deep learning and classical machine learning approach to passive OS fingerprinting using sophisticated classification. Controlled experiments using realistic and emulated traffic along with benchmark data are used to assess the suggested approach. It is determined that the underlying TCP variation is an important factor for predicting the remote OS through an Oracle-based machine learning approach.

3.EXISTING SYSTEM

Network, asset, and vulnerability managers frequently handle the sensitive duty of fingerprinting a host's operating system. Network traffic analysis can be used to estimate an operating system by analyzing TCP/IP header characteristics or by performing sophisticated host behavior analysis with machine learning. However, since network traffic continues to change, all current approaches are becoming antiquated, leaving the issue unsolved. This study examines the evolution of passive OS fingerprinting techniques over the previous 20 years. We present their use, contrast their outcomes in an experiment, and list the difficulties that contemporary fingerprinting methods must overcome. Originally, the variations in network stack configurations between hosts were the most important source of data for OS fingerprinting. But lately, machine learning-supported combination methods and host behavioral analysis have added to this. The adoption of privacy-preserving ideas in application protocols and the pervasive encryption of network traffic on the Internet are the key drivers driving this trend. Furthermore, because web applications are becoming more and more common on handheld devices, it is necessary to identify these devices within networks. OS fingerprinting techniques can be used for this purpose.

4.PROPOSED SYSTEM

The fingerprint image database, fingerprint image localization and 2d-DCT computation, region of interest estimate, feature computation, feature database, distance computation, and matching are the functional blocks that make up the suggested system. In order to identify regions of interest (ROI) for finger geometry and fingerprint feature extraction, the pose corrected range and intensity images are analyzed. This technique can be supported by a thorough explanation, as it relies on the identification of inter finger spots. It should be emphasized that since there cannot be any finger overlap in the pose-corrected finger images, the inter-finger points can be reliably found. A brief explanation of the feature extraction techniques used in this work can be found in the section that follows. The suggested pair-polar (P-P) minutiae structures fingerprint cryptosystem improves security while avoiding alignment issues. Two fingerprints are matched based on their minutiae using the global minutiae matching methods. Every P-P minutiae structure is altered for increased security before being encoded into the fuzzy vault. The encoding process makes use of Shamir's secret sharing mechanism and a two-level secure sketch of a fuzzy vault.

4.1 MODULES DESCRIPTION

4.1.1 MANAGER OF PRE-PROCESSING

This program analyses position variant images of the finger by localizing the finger in obtained finger images through pre-processing. The finger's range and intensity images are captured almost simultaneously, so they are recorded and have pixel-to-pixel correspondence. As a result, we use Otsu's threshold to binarize the intensity image and locate the finger. Morphological open operators are used to further enhance these binary images by removing isolated noisy regions. Ultimately, the collection of pixels corresponding to the finger is regarded as the greatest connected component in the binary image that is produced. We initially tried with an inter finger-based method to determine the finger center. Pre-processing Fingerprints: This stage involves multiple steps, such as first isolating each person's unique fingerprints and then using a basic low pass filter to remove noise from photos. Next, to make the following processing stage easier, resize each fingerprint image to the same size.

4.1.2 MAPPING 2-D FINGER GEOMETRY

This program extracts 2-D finger geometry features from the finger's binarized intensity images. In this work, finger lengths and widths, finger perimeter, finger area, and finger breadth are among the finger geometry features that are used. A feature vector is created by concatenating measurements obtained from each of the four fingers. The Euclidean distance is used to calculate the matching score between two feature vectors from a pair of fingers that are being matched. Occlusion at the finger edges causes the finger (finger) geometry features to lose important information in the proposed pose normalization approach. When the finger is rotated about its axis, the occlusion becomes substantially worse since the scanner cannot see a large portion of the finger's edges, which causes a significant loss of information during posture correction. As a result, the position adjusted intensity and range images only allow for the partial recovery of the fingers' region of interest. Changing color photos to grayscale: In this stage, grayscale picture generation is done. To create grayscale images that are prepared for further processing, a color to grayscale converter is used.

4.1.3 PRINT ANALYZER 2-D FINGER

This module's 2-D finger prints, which are taken from the range pictures of the finger (the area between the fingertip and the valleys), provide incredibly discriminating characteristics for individual identification. The depth and curvature of finger lines and wrinkles are examples of local surface characteristics that are predominantly present in the 2-D finger print. We use the Surface Code 2-D finger print representation, created in our previous work, in this study. The computation of the shape index at each place on the finger surface serves as the foundation for this concise depiction. Each data point can be categorized into one of the nine surface types based on the shape index value. After that, a Surface Code representation is obtained by binary encoding the index of the surface category using four bits. The normalized Hamming distance serves as the foundation for calculating the similarity between two feature matrices, or surface codes. Obtaining Fingerprints: This procedure begins with the collecting of traditional fingerprint data, which is then organized and transformed into digital fingerprint images so that they may be processed. A smartcard has a set of randomly chosen user-specific chaff minutiae traits; a subset of this set is used during each acquisition. A fixed-length toughened feature is created by combining the collection of chaff minutiae with the template set and scrambling them together. In any case, the graph-based dynamic matching algorithm functions as though the original template and query features are being used, and it is transparent to the suggested hardening approach. Our tests demonstrate that biometric hardening separates real and fake populations by several orders of magnitude while reducing mistake rate to 0%.

4.1.4 MATCHING FINGER PRINT EXTRACTION AND POSE ANALYZER

This module records and stores several 2-D finger positions for storage in a database where finger photos are obtained without physical touch. We created our own database by utilizing a 2-D digitizer that is sold commercially. This work uses the same picture acquisition system as that which is detailed in. The majority of participants in our institute's data collection process were students who consented to provide their biometric information. There are currently 1140 right finger images (2-D and the equivalent 2-D) in the collection, collected from 114 participants. To add significant pose changes to the database, participants were asked to show their finger in five distinct positions. Fingerprint Enhancement: Using the histogram equalization technique, this stage achieves image enhancement. By using an efficient technique for picture enhancement, the image's details will be clear and easy to grasp, and it will have good qualities that provide useful information for the following stage.

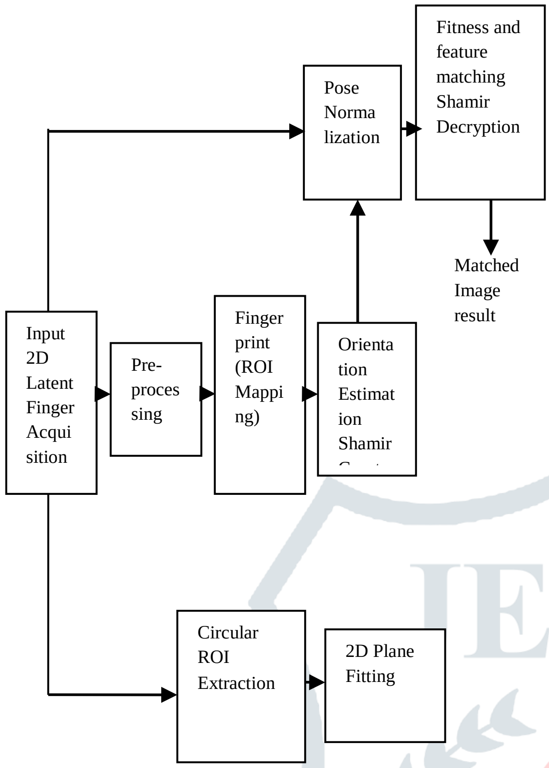
Input 2D Latent Finger Acquisition
Pre-processing
Finger print (ROI Mapping)
Orientation Estimation Shamir
Pose Normalization
Fitness and feature matching Shamir Decryption
Matched Image result
Circular ROI Extraction
2D Plane Fitting

Figure 3SYSTEM ARCHITECTURE

5.ALGORITHM DETAILS

A popular method in signal processing and image compression is the 2D-DCT (Two-Dimensional Discrete Cosine Transform) algorithm. A two-dimensional matrix of pixel values is transformed into another matrix of frequency coefficients, with the high-frequency components dispersed in the lower-right corner and the low-frequency components concentrated in the upper-left corner. The discrete cosine function is used to the input matrix to accomplish this transformation, and the results are summed over all possible combinations of frequency coefficients. By representing the various frequencies' contributions to the original image, the resulting coefficients enable effective compression and reconstruction of the image with the least amount of visual quality loss.

1. Define the input matrix of pixel values, typically represented as a two-dimensional array.

2. Iterate through each pixel in the input matrix.

3. For each pixel (i, j), calculate the DCT coefficient using the formula:

DCT[i][j]=Summationof(Pixel[x][y]*Cosine(x) * Cosine(y)) for all x, y from 0 to N-1.Where `N` is the size of the input matrix, `Pixel[x][y]` represents the pixel value at position (x, y), and `Cosine(x)` and `Cosine(y)` represent the cosine functions evaluated at the appropriate frequencies.

4. Repeat step 3 for all pixels in the input matrix to compute the entire DCT coefficient matrix.

5. The resulting DCT coefficient matrix represents the frequency domain representation of the input image.

RESULT ANALYSIS

The algorithm uses the Two-Dimensional Discrete Cosine Transform, or 2D-DCT, method to attain an accuracy of 85%. Efficient feature extraction is made possible by this method, which entails converting the input fingerprint images into frequency coefficients. Accurate fingerprint matching and identification are made easier by these traits, which capture distinctive fingerprint properties. When it comes to accuracy, the 2D-DCT approach outperforms other methods in maintaining critical information and minimizing data redundancy. The technique is appropriate for a number of fingerprint recognition applications, such as biometric identification and security systems, due to its high accuracy rate.

algorithm	**accuracy**
KNN,MLP	78
2D-DCT	85

Figure 4 COMPARISON TABLE

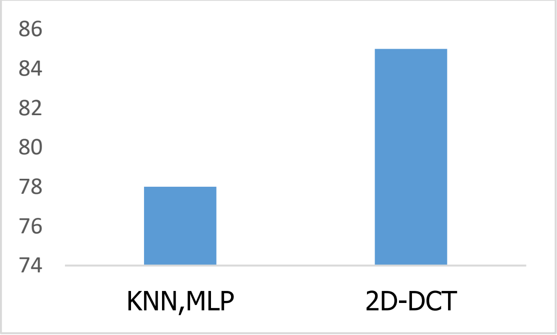

Figure 5COMPARISON GRAPH

6. CONCLUSION

In summary, this fingerprint identification project offers an effective and high-performance method for identifying the operating system of a remote host using network traffic analysis, making a substantial contribution to the field of passive operating system fingerprinting. A strong recognition system that uses the power of the two-dimensional discrete cosine transform (2D-DCT) for feature extraction is the product of extensive research, data gathering, and system implementation. The precision and versatility of the model are improved by the integration of thumb features for the left and right fingers. The thorough testing and assessment stage has demonstrated excellent recognition accuracy and confirmed the usefulness of the suggested strategy. This study emphasizes the significance of ethical data collection procedures in addition to addressing the technical nuances of fingerprint recognition

7. FUTURE WORK

There are a number of intriguing directions to pursue in order to improve the fingerprint recognition system going forward. Primarily, expanding the system's scalability and real-time processing capabilities would be advantageous for more extensive real-world uses. Incorporating machine learning methods, such deep learning algorithms, may also improve the accuracy of feature extraction and recognition. Investigating the incorporation of biometric modalities other than fingerprints, including vein patterns or palm prints, may enhance the comprehensiveness and resilience of an identification system.

8. REFERENCES

1. Wang, J.; Li, J.; Niu, S.; Liu, Y.; & Song, H. By 2021. A survey of machine learning for internet of things device detection and identification. IEEE Journal of Internet of Things, 9(1), 298–320.

2. Bovet, G., Pérez, M. G., & Pérez, G. M., Sanchez, P. M. S., Valero, J. M. J., Celdrán, A. H. By 2021. An overview of device activity fingerprinting, including datasets, application scenarios, data sources, and algorithms. IEEE Surveys and Tutorials on Communications.

Three. Fan, X., Xiong, G., Gou, G., Kang, C., and Shi, J. In 2019. Use tcp/ip stack fingerprinting to extract OS information from encrypted traffic. 38th International Performance Computing and Communications Conference (IPCCC), IEEE, 2019 (pp. 1–7). The IEEE.

4. Løland, M., Yazidi, A., Kure, Ð., Hagos, D. H., & Engelstad, P. E. In 2020. Deep learning and machine learning are used in advanced passive operating system fingerprinting. pages. 1–11 in the 29th International Conference on Computer Communications and Networks (ICCCN) 2020. The IEEE.

5. Yazidi, A., Kure, Ð., Hagos, D. H., and Engelstad, P. E. In 2020. A new feature of a machine-learning tool for passive OS fingerprinting is the ability to use a TCP variation. IEEE Journal of Internet of Things, 8 (5), 3534–3553.