International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 38 - 57, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1006

An Improved Security Architecture for Point-Of-Sale System

Terwase, Victor Sesugh
Department of Mathematics and Computer Science
Benue State University
Makurdi, Benue State, Nigeria

Aamo, Iorliam
Department of Mathematics and Computer Science
Benue State University
Makurdi, Benue State, Nigeria

Terwase, Aondona Isaac
Department of Mechanical Engineering
Heriot-Watt University
Edinburgh, United Kingdom

Abstract: Point-Of-Sale (POS) system has become ubiquitous and popular among micro and small-scale businesses such as retail stores, supermarkets, and other businesses for daily transactions especially in Nigeria. Among its numerous advantages such as better inventory management, simple invoicing, quick payment and others, it is fraught with a lot of security challenges or attacks some of which include: malware attacks, key logger attacks, and user identity attacks. The future of this promising technology looks bleak if these breaches or attacks are not identified and checked. This research proposes a detailed novel security architectural design identifying areas of possible breaches and possible solutions. It utilized KMeans clustering and KNearest Neighbour (KNN) algorithms on data collected from POS to classify the data generated and achieved an impressive result of 58.17% clustering separation and 99.51% accuracy classification of data points respectively.

Keywords: Point-of-Sale, Unsupervised Learning, Clustering, Attacks, Malware, Architecture, KMeans Clustering.

1. INTRODUCTION

The term Point of Sale (POS) is used to describe the technology used by a consumer to provide their payment information in exchange for a good or service[27]. POS technology has actually been around for many years with the first cash register dating back to 1879 [1]. However, it wasn't until the mid-70s that this technology was converted from mechanical to electrical form. Today's POS systems consist of many of the same components that are found in traditional information systems. One of the key differences between POS systems and other information systems is its key actors or stakeholders [4]. The primary key actors for today's POS systems are as follows: consumers, merchants, acquirer, issuers, card brand companies, payment processors, payment gateways, software vendors, and hardware vendors. Consumers are those people that use payment cards for the purchase of goods (mostly humans). Merchants are businesses that accept payment cards as a form of payment for goods and services. Merchants are also the implementers of the POS systems [14]. An acquirer, also referred to as an acquiring bank, handles authorization requests from payment processors and settles the transaction with the card issuer. Issuers provide the cards to consumers and maintain the payment card accounts. Card Brands also referred to as card networks (e.g. MasterCard, VisaNet), manage the overall process of authorization and settlement [3]. In the 21st century, the use of electronic payment (e-payment) systems to carryout financial transactions by micro, small and medium scale enterprises have taken center stage in developing and developed economies [5]. Since 2012, the use of point-of-sale terminal popularly called POS terminals to make financial payment and other bank transactions in Nigeria was introduced by the Central Bank of Nigeria (CBN) to promote its cashless policy with the aim of improving payment system. Statistical figures from the Nigerian Inter-Bank Settlement System [22] shows that as of 2018, the number of active POS across Nigeria was 164, 607. This has risen to about 686,577 with over 3 trillion worth of transactions as of March 2021 [22]. The attribution to this growth can be the growing acceptance of POS terminal for making payments and increase in network penetration in Nigerian [5]. Furthermore, the growth of cashless transactions and the impact of the Covid-19 pandemic is expected to drive the global electronic Point-of-Sale (POS) market to reach 2 million units by 2027. POS technology can be found in various businesses such as cafes, bars, hotels, hospitals, gas stations, retail stores, and saloons. Some POS systems are cloud-based, allowing for payments through mobile devices, while others come with added features such as appointment scheduling for saloons[2]. Additionally, many merchants offer POS financing

options, allowing customers to make large purchases and pay in installments through companies such as Affirm, Afterpay, and Klarna. Popular POS systems include. Square, TouchBistro, Poster POS, Vend with Vend being a combination of web and mobile.

According to the research conducted by[19], there are a number of POS system categories. A typical POS system is made up of several client computers connected through privately owned connection lines, such as the electronic data exchange (EDI), with local servers at one or more stores (Figure 1). The servers run all data processes for these on-line POS systems, while the client computers provide the user interface operations. Stemming from this strong, server-dependent design is the need to maintain the connection between the server and the clients throughout the processing of a sales transaction as disconnection will result in data loss and force the client(s) to suspend all transactions (sales/entries) until the link is re-established [8]. Subsequently, in this type of POS system, disconnection from the server can be a major hazard, and small business owners, who are vulnerable to the frustrations of their customers, may have to bear the expense of having an in-house server at each store location [19].

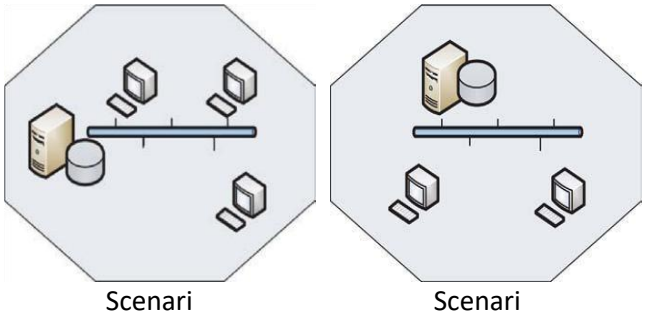

Figure 1 Local Client Server Model for POS system [19].

Moreso, another type of POS system is an off-line, batch-based POS system. In this system, all clients are capable of processing all transactions with their local data cache, and the processed transactions can be transmitted to the server periodically or on demand. For example, salespersons would upload the records of transactions in their handheld POS terminals (clients) when they go off duty. Since this type of off-line POS system is naturally immune to the hazard of being disconnected from the server and does not collect real-time information from its clients, the system will be adequate for certain business environments in which there is weak or limited network connectivity. It is important to note here that, the Electronic Payment System (EPS) functionality of the POS was added to enhance the robustness and easier business transactions to clients.

In a recent article published by the Information Technology Magazine[24], payment terminal malware has stolen $3.3m of worth credit card numbers in the United States alone. In developing countries, there are no concrete data to illustrate the quantum loss in monetary terms or otherwise as a result of malware at the terminals of the POS. The article further pointed out that cybercriminals have used two strains of point-of-sale (POS) malware to steal the details of more than 167,000.00 credit cards from payment terminals which when sold at the underground market is worth $3.3m[18].

Due to advancements in technology, cost reduction efforts, the desire to improve customer satisfaction, and keeping up with global banking trends, the use of electronic devices such as ATMs, POS terminals, and mobile phones for electronic transactions has become widespread in the Nigerian banking industry[6]. These transactions can now be conducted through online platforms, ATMs, POS systems, and mobile phones, among others. This new method of conducting banking business is referred to as Alternative Banking Channels (ABCs) [3]. The ABCs provide a variety of financial services, including cash withdrawals, fund transfers, cash deposits, payment of utility and credit card bills, request for checkbooks, and other financial inquiries. Most transactions on these ABCs are performed with a card, while some require card information.

Table 1 Types of Fraud with Frequencies [23, 22].

YEAR	CardPresent Fraud (ATM and PoS)	Card Not Present Fraud (online and Web)
2012	1539	314
2013	1739	316
2014	7181	1271
2015	8039	1471
2016	11180	3374

International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 38 - 57, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1006

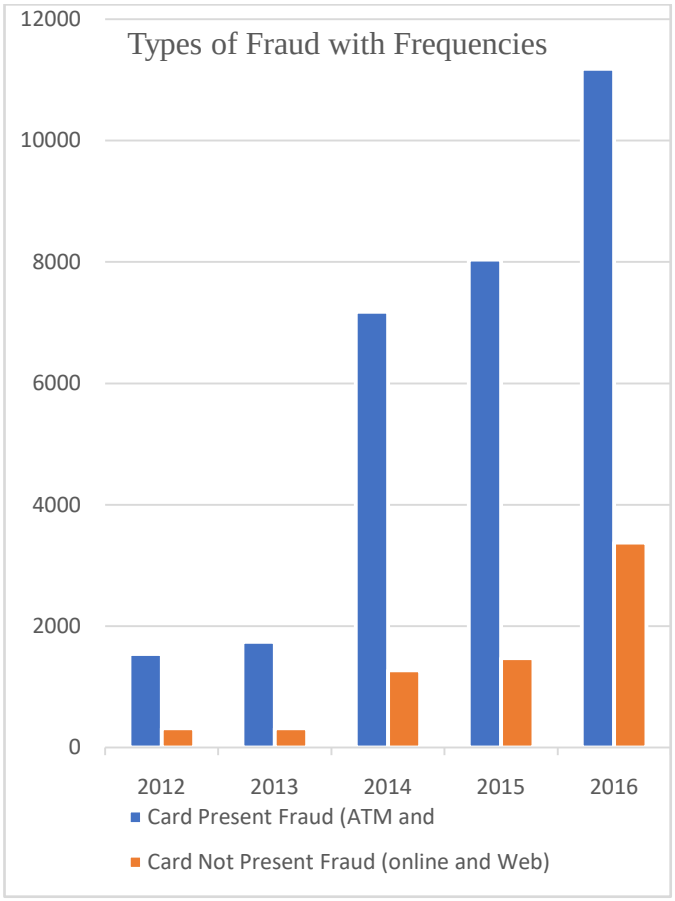

Figure 2 Types of Fraud with Frequencies (card present and card not present).

The number of reported fraud cases on Alternative Banking Channels (ABCs) has been steadily increasing. As shown in Table 1, the number of reported frauds on ATMs and POS terminals rose from 1539 in 2012 to 11,180 in 2016, representing a growth of 626% in just five years. Similarly, the number of reported frauds on online and web platforms increased from 314 in 2012 to 3,374 in 2016, reflecting a growth of 974% in just five years. According to [2], the losses due to fraud through POS terminals increased from N5.8 million in 2013 to N157.6 million in 2014, while mobile banking fraud losses rose from N6.8 million in 2013 to N13.3 million. Meanwhile, the losses due to fraud through ATMs and online banking declined from N1.242 billion to N0.5 billion and from N3.196 billion to N0.875 billion, respectively, over the same period. [23] Annual Report indicates that the actual fraud losses on ATMs, internet banking, POS, and web platforms were N464.5 million, N320.7 million, N243.3 million, and N83.8 million, respectively. The increasing number of fraud incidents on these ABCs may lead to a further loss of public confidence in these technologies, which were intended to provide convenience and comfort in banking and business transactions. Customers are losing trust and confidence in the banking system due to rampant frauds.

This calls for great concern considering the huge amount of monies loss due to security breaches at the POS terminals. Point of sale security is the prevention of unauthorized access to electronic payment systems by individuals who are typically looking to steal customers' personal details such as credit card information [20].Point-of-sale security (POS security) is also the study of vulnerabilities in retail checkout points and prevention of access by unauthorized parties looking to steal customer and payment card details from them. The purpose of POS security is creating a safe environment for customer transactions [11].

POS security aim to create a safe environment for customers to complete their purchases and transactions, and it is an important measure for fostering trust with today's business client or consumers. Understanding the areas where card data is vulnerable provides the area to look at some of the attack methods that have been used by hackers for intercepting payment card data within the POS system [25][10].

By integrating machine learning models into real-time monitoring systems, companies can identify fraudulent transactions as soon as they happen and take prompt action to stop additional losses [4].

Through the use of machine learning techniques and ongoing adaptation to changing fraud tendencies, firms may proficiently identify and avert point-of-sale fraud, safeguarding their funds and reputation.

2. RELATED WORKS AND BACKGROUND

To provide context for our review and analysis, it is important to understand that an Electronic Payment System (EPS) is a separate function from the typical POS function, although the EPS and POS system could be collocated on the same machine. In general, the EPS performs all the payment processing while the POS system is the tool used by the Cashier or Consumer (e.g self-checkout kiosk for the consumer) [14][7]. When looking at the payment systems, it is importantto follow the path the payment card data takes because the data is what is valuable to a hacker. The payment data enters the system via the POI device and then makes its way through processing – as seen in the diagram below. In a store EPS deployment model, the POS and EPS functions are located on separate Machines.

International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 38 - 57, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1006

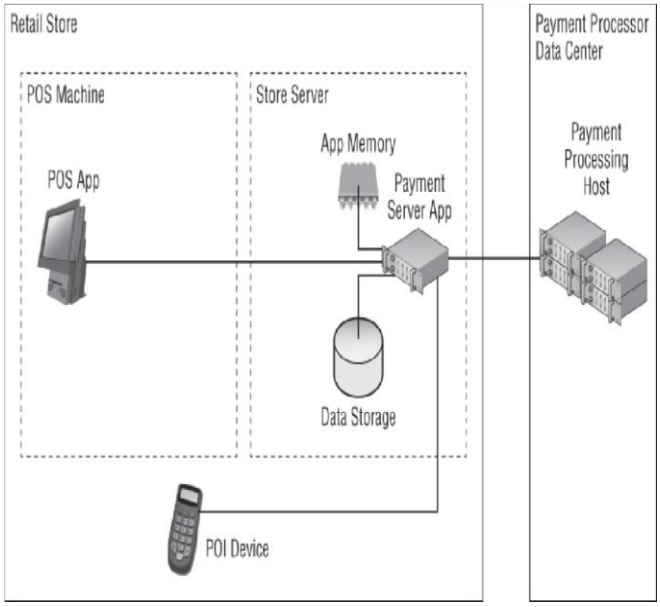

Figure 3 Point of Sale (POS) EPS Deployment Model [14].

Essentially, the EPS is serving as a "middle-man", which prevents any sensitive data from entering the actual POS system. As seen above Figure 3. The POI device connects directly to the EPS (i.e.Store Server) instead of the POS machine. In a POS/ EPS deployment model, the POI function and the EPS function are both are connected on the same system.

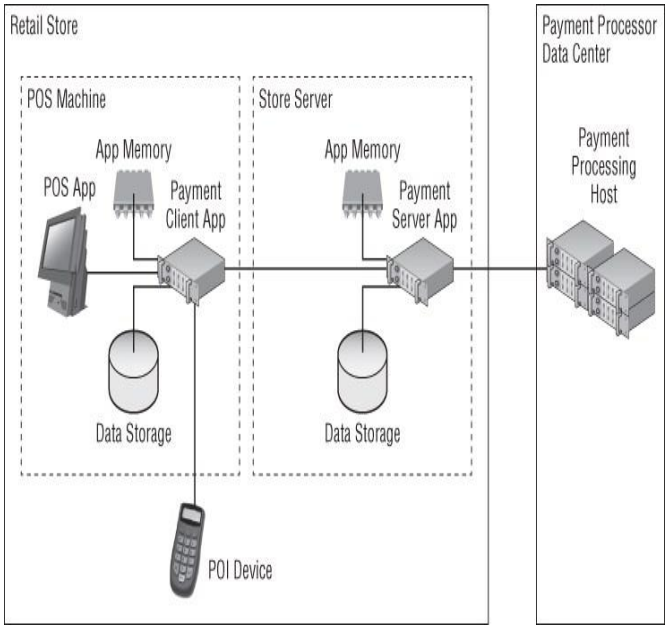

Figure 4Hybrid/POS Store Deployment Model [14].

This places the payment processing function on the actual POS machine. Thus, the POS machine is exposed to sensitive data in this model. In a hybrid/POS store deployment model (Figure 4), the EPS functions are broken up across multiple systems. In this model, multiple machines are exposed to sensitive data creating multiple targets of opportunity for the bad guys[14]. Furthermore, the internal network of these POS deployment models needs to be secured from the bad actors otherwise, this will make the model vulnerable[27].

The deployment models analyzed and depicted so far represent the POS systems seen in most retail stores. However, it's worth mentioning that other deployment models don't fit into the categories above such as gas station payment systems and mobile payments (e.g. NFC). The primary differences between these models and the ones mentioned are that there are a few different pieces of software (e.g. mobile apps) and hardware (e.g. mobile phone, fueling pump)[16].

POS breaches and occupational fraud (fraud committed by an insider) remains on the increase and a lucrative endeavor for fraudsters[21]. Unfortunately, the current retail point-of-sale payment system architecturesarchitecture are fraught with many security challenges and effectively detecting these security challengesis an issue that urgently need to be addressed in other to realize the full potential in the POS payment system. There are several POS systems in Nigeria, and it is reported that there exist several fraudulent activities being perpetrated by some fraudulent POS operators [5]. Some of these criminal activities include: keystroke logging, debiting a customer's account without his/her knowledge, password theft, malware attacks, physical tampering or skimming, identity theft from fraudsters or even dubious persons who use POS in retail supermarkets or businesses just to mention a few Several measures and other architectures have been developed to check some of these crimes/data breaches in POS systems but have not been enough. Most of these attacks or crimes go undetected, this research is motivated by identifying and using effective ways of detecting fraudulent transactions using data from point-of-sale system[13][15].

In [28], "A Novel Approach of Unprivileged Keylogger Detection" focuses on identifying unprivileged userspace keyloggers, which are software applications created to secretly record and capture user keystrokes. The authors stress the need to protect user input on computers and the risks that keyloggers may present, especially in systems like online banking where sensitive data is submitted.The authors draw attention to the fact that the majority of keyloggers used today run in userspace mode, which doesn't need special rights to run. They suggest a method based on detection techniques that compares the I/O activity of processes with simulated user activity in order to detect userspace keyloggers. Keyloggers are said to require a substantial amount of I/O activities in

order to record keystrokes, and this pattern can be used for detection.n.

The paper conducts a thorough literature review, outlining the background of keyloggers, how to categorize them, and what research has already been done in the area. As instances of recorded incidents when keyloggers were used maliciously, it also discusses the potential risks and losses brought on by keyloggers[12].

The authors describe their research technique, which include examining different keyloggers' behaviour both with and without simulated user interaction. They offer C++ code samples to demonstrate their methodology and how they altered API calls to find keylogger activity.

The findings section displays their proposal's capacity for detection.

[17] in his research, gives an overview of the rapid developments in mobile telecommunication and handset technology, which have improved user experiences and led to the emergence of powerful smartphones that can mimic desktop computer features.Mobile payment (m-payment) systems have emerged as a result of the advent of products sales over the internet via mobile devices, which has increased the demand for secure payment methods[9]. The researchers emphasized the need for standalone mobile point of sale (POS) apps that cover safe financial transactions is highlighted in this study. Figure 5 shows the system architecture of the POS system we have in mind. In this architecture, the administrative user enters user data into the local database using a stationary computer. Employees use mobile devices to access the POS system, as do managers who use smartphones or tablets. Employees may have additional POS equipment in some circumstances to handle customer payments. Employee mobile devices must support NFC in order to accept mobile payments from customers using NFC-enabled devices. Certificates are issued to system entities by the Certificate Authority (CA) server. Because of the way our system is set up, users with SAFE accounts can pay invoices directly from their accounts. To manage customer payments, the system is additionally connected to the bank's IT server.

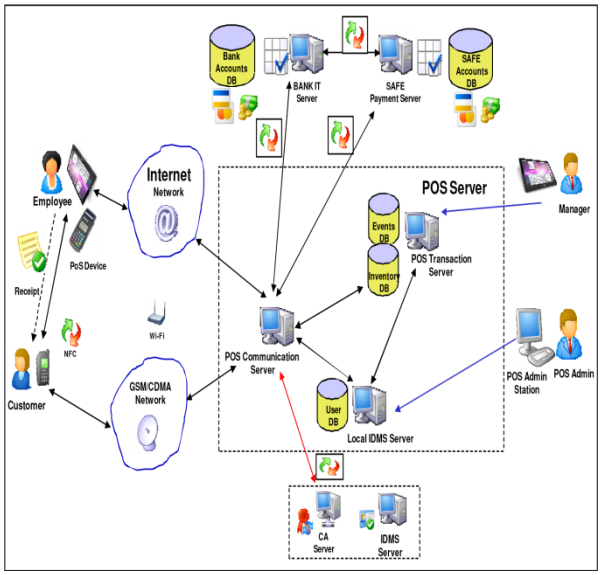

Figure 5System Architecture of the Mobile POS System[17]

Figure 5 shows the internal organization of the POS server that we created. The POS server includes a number of essential parts, such as:

Inventory IDMS (Identity Management System) database

Administration Service Security Manager Transaction Manager Communication Manager A programming language interface (API)

API Manager Service

API for Employee Service

Client/Customer Service API

The Manager application, Employee application, and Customer application are three mobile client applications that connect to the system via their respective APIs.

[16], provided an investigation into the use of various machine learning methods for the identification of "server rotating bill item" fraud in a dataset from a restaurant point-of-sale (POS) system. The effectiveness of several fraud detection algorithms is evaluated in the study, along with the effects of engineering features and artificial features on the models' performance. Here is a summary of the paper's main points:

The research highlighted the significance of feature engineering and model selection in generating accurate findings and shows the promise of machine learning

techniques for detecting insider fraud in restaurant point-of-sale data. Decision trees (RandomForest), probabilistic classifiers (NaiveBayes), artificial neural networks (NeuralNet), k-nearest neighbour, linear/kernel-based classifiers (Support Vector Machine), and Adaboost are a few examples of machine learning techniques they employed on data from various restaurants.

3METHODOLOGY

3.1 Overview

This research employs qualitative and quantitative method in data analysis. Based on the literature reviews designed a conceptual architecture and proffer deep analysis on areas of likely breaches and possible solutions. The research assumes that the data contains transactions that are fraudulent and employs unsupervised and supervised machine learning to cluster and obtain insights into the datapoint which could help to detect fraudulent pattern. Unsupervised and supervised machine learning techniquessuch as kMeans clustering and KNearest Neighbour to get some insight and calculate the accuracy of the predictions. Some important metrics were used to evaluate our model.

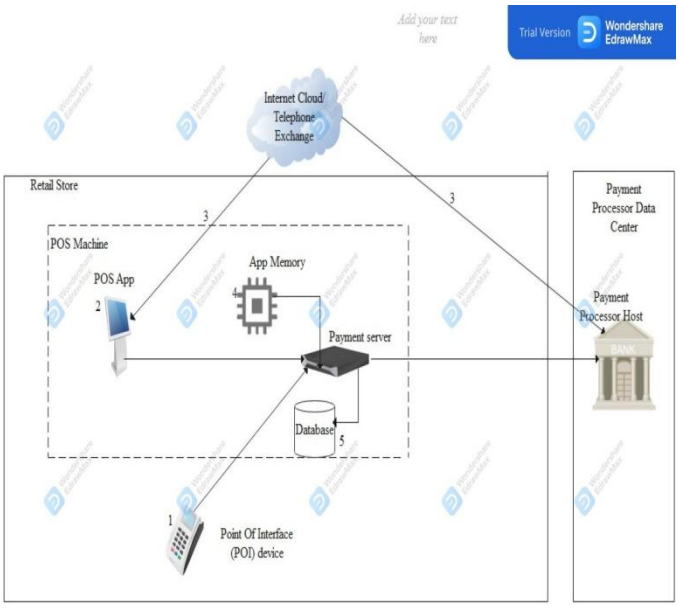

Figure 6Proposed Conceptual Security Architectural Design of POS System.

In our architecture in Figure 6, we try to ensure that we separate the POI terminal from the POS machine(computer) and separate the Point-of-Sale's internal network from the public network. To enhance further security of the model's network, we propose state of the art network security encryption techniques.

POS breach phases don't necessarily have to happen in any particular order but generally there is some consistency in the methodology.

Attacks on Terminals: attacks on the terminals can come from Id_theft, keystroke logger, Skimmers, Firmware, inserted hardware, malware assaults, physical tampering., Internal Network traffic sniffing.

1. Attacks on terminal (1): It has been observed based on experience that, about 3% of point-of-sale (POS) operators especially retail businesses devise devious ways to exploit systematic weaknesses to steal cash, credit card data and stock, either single-handedly or in collusion with other bad actors at the terminal. These fraud techniques hold until detected at a point of historical recovery. Before considering the layers of internal control and data analytics designed to prevent or detect retail POS fraud, here are some outlines of various schemes by which fraud can be perpetrated by POS operators and their collaborators, especially at the terminals.

(a) Manipulating Voids: An employee voids a valid transaction and keeps the money for themself in this type of POS theft. The staff at the grocery store or retail store gives the item to the consumer, but voids it off the bill and keeps the money.

(b) False Refunds: Employees steal real cash transactions in this particular blatant POS theft or attack, returning the money to their personal account after the consumer departs. It is possible for certain retail staff, such as cashiers, to reimburse credit card transactions to their personal credit card account.

1. Public networks are vulnerable if the device does not adhere to accepted modern security standards for credit card payments (PCI DSS) or if there are system vulnerabilities, such as the encryption key being exposed.
2. RAM scraping memory attacks: While a card is swiped or inserted, it details are transiently stored inside the terminal's memory while being transmitted to the payment processor. This presents a brief period for malware on terminals or charged processor reminiscence to copy vital card details.
3. SQL injection through public network to the database.

All the aforementioned attacks at the terminal by the POS employee can be classified under human behavior. Now we can now look at some solutions to these breaches:

1. i. A clear zero tolerance anti-fraud policy that spells out the repercussions for non-compliance, preventing any staff members who are subsequently discovered scamming the company from using ignorance as a defense.

ii. To guarantee individual accountability for every transaction, a mandatory POS login at the beginning of every shift and a mandatory logout at the conclusion are required. Operator logon code sharing should be strictly prohibited.

iii. Cash safes and point-of-sale systems should be placed so that operator activity is constantly visible and within the visual range of CCTV cameras. Printed dockets with the till operator's ID routinely provided to
the customer so that individual transactions can be traced back to the operator.

iv. There should be Job separation between end-of-day totaling, banking, start-of-day float, refunds, and point-of-sale operations.

v. POS operator awareness that there is a shopper regime in place by which the above controls are randomly checked by anonymous members of management and audit.

A keystroke logger attack is the term for a malevolent computer program that surreptitiously logs the keystrokes that the point-of-sale (POS) user makes. In point of sale (POS) systems, keystroke tracking poses a unique problem to the security manager. Key loggers are devices—either hardware or software—that record characters from a keyboard and send them to a connected computer or point-of-sale system. Keystroke logging have both ethical and unethical application. Among some of the ethical application include. (1) Quality assurance testers analyzing sources of system error (2) System developers and analyst user interaction with systems (3) Employee monitoring (4) Law enforcement or investigators looking for evidence against a criminal suspect.

There are four sorts of key loggers: software, hardware, wireless, and acoustic. Their mode of operation differs on how they capture information. When using hardware or software keyboard loggers, the compromised system stores the log files. Keystroke data is recorded by software key loggers while it is transferred between the operating system and the computer interface. They can be developed as conventional apps or as kernel-based keystroke logging apps that use a hooking technique to record data from the keyboard. Keystroke loggers in both application and kernel software capture keyboard input, write an encrypted copy to a local log file, and then send the data to the operating system.

A hardware key logger is essentially a circuit that is positioned in the space between the computer and keyboard. Hardware key logger is connected directly to the POS keyboard or interface. Once, the key logger is connected, it immediately begins keystroke collection. Character and control code data are captured by the logger's CPU and written to the onboard memory.

Numerous methods have been put forth to identify and counteract this attack. One example of this is the usage of firewalls and anti-malware software on POS system terminals, which occasionally is insufficient to thwart system attacks. Following the criteria of the Multifactor Protection System, One example of this strategy in action is the use of firewalls and intrusion detection systems (IDS) to protect point-of-sale (POS) terminals against malware. Using the Host Based Intrusion Detection System (HIDS) is another method. An intrusion detection system, or HIDS, logs suspicious or malicious activity and analyses traffic on the computer it is installed on. (cybersecurity.att.com). HIDS resides in a single hosting system monitoring and reporting on the system's configuration and activity. This added level of security protection ensures malware that passes the firewall does not leave the system vulnerable to attack. HIDS has many facets such as signature detection, anomaly detection and stateful protocol analysis detection to protect against malicious threats.

Secondly, another type of attack that might occur at the POS terminal is the skimming attack. Skimming is the illegal use of a rogue physical device, which frequently appears as a component of a POS terminal or other device, to transfer and acquire important data for the malevolent use of a skimmer. The POS system's integrity is compromised via skimming. Skimming devices can be rather complex, tiny, similar to a tiny chip and hard to spot because they frequently blend in with the POS system terminal. Skimmers are capable of recording the data embedded on the magnetic stripe of debit/credit card data as they are inserted into the payment terminals of the POS. The skimmer is installed as an overlay that blends in or is identical to a genuine terminal, or it is concealed inside the terminal's card reader.

2. Internal Network traffic sniffing: At this stage, sniffing may happen, particularly if the adversary (attacker) uses sniffing tools to penetrate the network. When it comes to retail point-of-sale systems, internal networks are used before they are connected to public networks.

The data on a credit or debit card is read and sent over several networks when a customer pays with a swipe at a point-of-sale system, ultimately arriving at the payment processor for the POS retailer. When data moves over these networks, it needs to be secured. Secure Socket Layer (SSL) or other network level encryption is required to secure data on public networks. Credit card numbers and other sensitive data are not needed to be encrypted within internal networks and systems unless they are being stored. Albert Gonzalez exploited this in 2005 by breaking into shop networks using network sniffing tools and collecting millions of credit card numbers as they travelled via internal networks. This type of challenge can be mitigated by the use of network level encryption

within the POS retailer's internal network. Also, the use of point-to-point encryption protects data while undergoing processing.

3. Public network attack: as previously mentioned, data from point-of-sale systems travels via multiple networks prior to reaching the payment processor (bank). If the network does not adhere to the accepted standard for safety procedure for card payment, such as PCI DSS, then critical data or credit/debit card credentials may be vulnerable to attack. Sniffing (using a packet analyzer) is one method of attack.

A denial of service (DoS) assault aims to overwhelm system resources so that no one else can use it, rendering the system unusable or severely slowing down the system or public network for authorized users. The goal of a denial-of-service (DoS) attack could be to stop a user in a point-of-sale system from connecting outside of the network. A DoS attack may also target an entire organization, to either prevent outgoing traffic or to prevent incoming traffic to certain network services, such as the organization's webpage or data.

DoS attacks have become very common on the Internet. Deliberate or unintentional DoS attacks are both possible. When an unapproved user intentionally overloads a resource, it becomes a deliberate DOS. It is caused accidentally when an authorized user unintentionally does something that causes resources to become unavailable. Most DoS attacks rely upon weaknesses in the TCP/IP protocols. This research work looks at some of the DoS attacks that could occur in an electronic payment system network:

1. SYN Flood Attack

This kind of attack happens when a host receives so many Synchronization (SYN) packets requesting incomplete connections that it is unable to handle valid requests for connections.

A three-way handshake is the series of messages sent by the client and server during an effort to establish a TCP connection between the client and server. Actually, the client system starts by communicating with the server via a SYN (synchronization) message. Subsequently, the server sends the client a SYN-ACK (acknowledgment) message in response to the SYN message. The client then finishes establishing the connection by responding with an ACK message. Next, a connection is established between the client and the server so that the client and the server can exchange service-specific data.

The point at which the server system has acknowledged the client (SYNACK) but has not yet received the final ACK message is where abuse might occur. This is referred to as a half-opened connection. The server includes a built-in data structure that lists all pending connections in its system memory. This data structure is of a particular size, and it can be made to overflow by intentionally creating too many partially opened connections.

With IP spoofing, creating a partially opened connection is a simple task. The source address of the SYN packets that the attacker's system seems to be sending to the victim's server is really spoofing a system that is not currently connected to the network. This implies that the final ACK message is never sent to the victim server. Because the source address is spoofed, there is no way to determine the identity of the true attacker when the packet arrives at the victim's system.

2. Teardrop Attack

The method used to reassemble fragmented IP packets is a vulnerability that teardrop attacks target. Fragmentation is required when IP datagrams are larger than the maximum transmission unit of a network segment that the datagrams must pass through. Each fractured packet's IP header contains an offset that indicates where the fragment fell within the original, unfragmented packet, allowing packets to be correctly reassembled at the receiving end. In a Teardrop attack, packet fragments are deliberately forged with overlapping offset fields, causing the host to hang or crash when it tries to reassemble them. Figure 7 shows that the second fragment packet purports to begin 20 bytes earlier (at 800) than the first fragment packet ends (at 820). The offset of fragment packet 2 is not in accord with the packet length of fragment packet 1. This discrepancy can cause some systems to crash during the reassembly attempt.

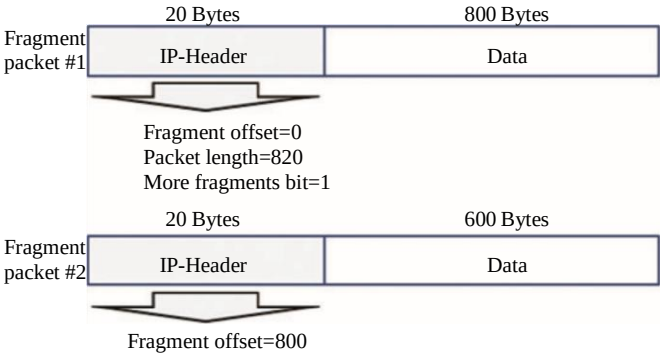

Figure 7**A** Tear Drop Attack.

3. UDP Flood Attack

Since UDP is a connectionless protocol, data transfer can occur without the need to establish a connection. When a hacker sends a UDP packet to any random port on the target system, it could result in a UDP flood attack. The victim system will identify the program that is waiting on the target UDP port upon receiving a UDP packet. Two scenarios could occur. The victim host will create an ICMP packet with a destination unreachable to the forged source address if there isn't an application listening on the port (closed UDP port). However, if there is an application running on the destination UDP port, then the application will handle the UDP packet. In both cases,

if enough UDP packets are delivered to destination UDP ports, the victim host or application may slow down or go down.

We now analyze how the use of biometrics can mitigate against some of these public network attacks.

The term “biometric” comes from the Greek words “bio” (life) and “metric” (to measure). Biometrics refer to technologies used for measuring and analyzing a person’s unique characteristics. In the security analysis research experiments results demonstrated clearly that tested biometric readers are very vulnerable to common Denial of Service (DoS) attacks, and their recognition performances significantly deteriorate just after launching the attacks.

Biometric devices can be easily crashed or disconnected from the network by common DoS attacks. The following lists some basic security considerations that should be taken into consideration when designing secure biometric readers to limit the effect of DoS attacks:

(a)The biometric reader’s user interface should allow the filtering of network packets, such as blocking all incoming ping requests.

(b). Network traffic with high-speed rate targeting the biometric reader should be denied from reaching the kernel of the reader. This would allow protecting the reader from many common DoS flood attacks, such as SYN flood attack.

(c) The ARP cache of the biometric reader should be static, so that malicious ARP packets cannot update its contents with fake IP/MAC entries. This would allow protecting the reader from DoS attacks based on ARP cache poisoning attack.

One way to mitigate this type of attack is the use of double encryption data (Encryption data and use of SSL).

4. RAM scraping memory attack: This is the type of attack where the terminals of POS system is attacked with a malware to copy card data which is then transmitted to the attacker. When a card is swiped, it’s details are briefly stored at the terminal’s memory while being transmitted to the payment processor. This provides a brief window on the terminal to copy the card data which is then transmitted to the attacker. This technique is referred to as RAM-scraping. RAM-scraping malware is used to collect numbers as they are read into the POS terminal’s memory. Any gathered data is locally stored in a file until it is transferred to the attacker’s computer, hopping through internal networks until it reaches a system designated by the hacker that has access to external network. In this type of attack, the attacker may hijack an internal system to act as a staging server. The adversary (attacker) may attempt to identify a server that communicates with the POS system often and leverage on a normal communication to avoid detection. Any data collected by the RAM-scrapping malware will be sent to the staging server where it is collected over some time and transmitted to the attacker. The data is transferred through internal servers before finally arriving at a hacked external FTP server.

Credit cards also contain a three- to four-digit number printed or embossed on either the front or back side called the CVV, “Card Verification Number (CVN),” “Card Security Code (CSC),” “Card Validation Code (CVC2),” or some other similar term depending on the credit-card-issuing institutions. These institutions have different names for this number but it is a security verification feature used in “card-not-present” transactions (e.g., made via telephone, mail order, online, etc.). Merchants cannot physically verify if cards are present for transactions. It is important to note that by design, this number is not stored in Tracks 1 and 2 and without it, an exact counterfeit credit card cannot be created.

The Primary Account Number (PAN) format, defined in ISO/IEC 7812, is commonly 16 digits long but can reach up to 19 digits and has the following format: IIII-IIAA-AAAA-AAAC.

The first six digits are known as the “Issuer Identification Number (IIN).” In a credit card processing, the length of individual account can reach up to 12 digit the final digit is a check which is calculated using the Lunn Algorithm. It has been identified that POS RAM scrapers generally use regular expression (regex) matches to search for and harvest Tracks 1 and 2 credit card data from the process memory space in the RAM. The complexity of the regex determines how it can correctly/incorrectly capture non-essential data from the RAM in addition to valid card data. A well-defined regex will return clean results but may be computationally more expensive compared with a looser one. If the hacker’s goal is to quickly capture data from the RAM, efficiency is more important than quality. To circumvent bad data problems, some PoS RAM scrapers implement Luhn validation to check the card data harvested prior to exfiltration.

Payment Card Industry (PCI) compliance for validating ex-filtrated data offline before selling it in underground forums.

The remedy from some of the loopholes exploited by POS Ram Scrappers can be remedied by applying the PCI Data Security (PCI DSS) framework in addition to other security measures.

PCI Data Security Standard (PCI DSS) refers to a set of requirements designed to ensure that all companies that process, store, or transmit credit card information maintain a secure environment.

PCI DSS does not offer new secure technologies to protect electronic payment systems but provides requirements to build up layers of security control around existing ones. PCI DSS v1.0 was published in December 2004, long after electronic payment systems were developed and deployed worldwide. At this point, defining, developing, and deploying a brand-new secure technology standard for payment cards would be extremely expensive.

PCI DSS has the following major requirements:

(1) Install and maintain a firewall configuration to protect cardholder data.

(2) Do not use vendor-supplied defaults for system passwords and other security parameters.

(3) Protect stored cardholder data.

(4) Encrypt cardholder data when transmitted across open, public networks.

(5) Protect all systems against malware and regularly update anti-malware solutions.

(6) Develop and maintain secure systems and applications.

(7) Restrict access to cardholder data on a need-to-know basis.

(8) Identify and authenticate access to system components.

5. SQL Injection through public network: As the adversary gain access to the public network server and traverse to the payment processor's network, the attacker may exploit vulnerabilities in the payment processor's network or exploit other techniques such as SQL Injection to gain access to the payment processor's database.

In recent times, the internet plays a very vital role in web applications, various Web application are signed up and make transaction through the retail point of sale terminal. Some of the POS terminals are used in sending mail, acknowledgement of goods purchase and other internet related activities thereby making the data stored on the internet to become huge in size and more valuable. Generally, some of these web applications that run on POS terminals consist of front-end, database back-end. Database is central to the web application, it stores those necessary data including users' names, passwords, various statistics, financial information and so on.

Structured Query Language (SQL) provide a way to manipulate data and change a database structure. SQL injection attacks allow hackers get access to vital information without authorization and authentication, thereby causing serious harm to the web application. SQL injection vulnerabilities provide an entrance for hacker to execute SQL statements on database. Therefore, the hacker can harvest any sensitive information and even can destroy the database. Different measures exist to mitigate against SQL injection but there are no standard one yet established. SQL injection attacks destroy the confidentiality, integrity and availability of the target system completely. The cost of SQL injection attacks varies with the value of information stored in the database. In 2009, Kaspersky was hacked through SQL injection and a lot of data was leaked including users, activation codes, lists of bugs, admins. This event not only caused economic loss owing to the stolen activation code but also affected its reputation as a security company seriously. In the case of electronic payment system like the POS, the monetary cost could be very huge and devastating to the victim.

In general, SQL injection attacks can cause the following effects:

- Data leakage
- Data modification
- Hacker getting full control of the database
- Hacker getting complete control of the host system

There are different attack methods against SQL injection vulnerabilities some of these attacks include:

1.Tautologies

This attack stems from the vulnerability which displays all query records on the web page of the front end in a web application. In this type of attack, all records of current table can be gotten through constructing query statements with where clause that always return true.

2. Union queries

Due to the fact that the union keyword requires two query statements having equal number of fields, the number of fields of the former query statements must be guessed by observing the error messages generated by

order by 'keyword. This attack is against the vulnerability which can display more than one query records and SQL error messages on the web page. This method can get additional information by concatenating the results of malicious query statements behind original query results, for instance, the name of the database and records of other tables and so on.

3. Boolean-based

A generic web page will be displayed instead of a database error message when there is an error in query statements. This page will also be shown when the query result is null. In this type of scenario, the error-base exploitation will not work in this situation. The Boolean-base exploitation technique which constructs a series of Boolean queries with some special built-in function (ex. Ascii(), length(), substring()) against the server can work perfectly because of the different responses to WHERE clause. When the where clause returns true the back-end will display some query result. In the event that the where clause return false, nothing will be displayed. By this method, the attacker can infer some useful information through the response.

4. RESEARCH METHODS AND MATERIALS

The dataset used is unlabeled and to achieve one of objectives in this research, we used an unsupervised machine learning called KMeans clustering. We employed KMeans clustering to find some shared features with collection of observations with similar characteristics. KMeans Clustering is comparatively fast, simple compared to other algorithms and preferable especially when dealing with very large dataset.

However, this algorithm employed is relatively sensitive to large outliers and the seed i.e the starting condition that is used to initialize the algorithm. Also, as the dimension increases, a distance-based similarity measure converges to a constant value between any given samples. We can overcome this by using principal component analysis (PCA) on the feature data to modify the clustering algorithm however, as observed in our dataset, the number of dimensions is relatively small. From our observation of the dataset, all our features are numeric and to achieve the objective of analyzing and discovering hidden pattern patterns from our point-of-sale data, we need to understand the main steps and logic in our program.

In other to analyze POS dataset to gain insight and detect fraudulent patterns, we employ the use of unsupervised machine learning models: KMeans Clustering, KNearest Neighbor (KNN). This is because our POS dataset is without label and we try to mine or extract patterns that would provide insight from the dataset which can aid segregate and classify fraudulent datapoints with greater degree of accuracy. Some of the research tools used includes:

Software:

Visual Studio Code or Google Collaboratory, Microsoft Excel.

Libraries:

Pandas, Numpy, Matplotlib, Seaborn, Sklearn, Keras, TensorFlow.

Hardware:

Configuration:

HP Pavilion DV 6, Intel Core I5, 600GB HDD, 2.3GHz Dual Core Processor, 8Gb RAM, Windows 10 Operating System.

Programming Language:

Python Programming

4.1 Dataset Collection and Description

This research was conducted using a combination of literature review. The literature review involved a comprehensive analysis of existing research on POS system security, including studies on security breaches, vulnerabilities, and countermeasures. This helped to identify the key security challenges faced by POS systems. The study also involved an in-depth analysis of the security practices and systems used by three large retailers. Data was collected through a supermarket data for a period of Four Months [26]. This helped to validate the findings of the literature review and to gather practical insights into the challenges faced by POS systems in real-world settings.

The data used was retrieved from retail supermarket checkout/POS system logs and cashier operations stored in XML files, which contained various low-level transactional data [26]. Once extracted, it was aggregated into six CSV files with the most important information about (i) transactions; and (ii) cashier operations, refer to Table 1, respectively. The data concerns retail operations in a grocery supermarket, equipped with manned (service) and self-service checkout.

Table 1 Transactions Data: Fields, Data types and Descriptions.

Field	Type	Description
WorkstationGroupID	Integer	Type of checkout: service, self-service
TranID	Numeric	Transaction ID (date, store ID, checkout ID, sequence no.)
BeginDateTime	Date/Time	Date and time of transaction start
EndDateTime	Date/Time	Date and time of transaction end
OperatorID	Integer	Unique cashier ID
TranTime	Integer	Transaction time in seconds
BreakTime	Integer	Break (including idle) time in seconds
ArtNum	Integer	Number of items, i.e., basket size
TNcash	True/False	Cash payment flag (true when transaction paid in cash)
TNcard	True/False	Card payment flag (true when transaction paid by a card)
Amount	Numeric	Transaction value
WorkstationGroupID	Integer	Type of checkout: service, self-service

WorkstationGroupID: This feature refers to the group or location of the workstation where the transaction was carried out. It is an integer value.

TranID: This feature refers to the unique identifier of the transaction. It is a float value.

BeginDateTime: This feature refers to the date and time when the transaction began. It is a string (object) value.

EndDateTime: This feature refers to the date and time when the transaction ended. It is also a string (object) value.

TranTime: This feature refers to the duration of the transaction in seconds. It is an integer value.

BreakTime: This feature refers to the duration of the break taken during the transaction, if any, in seconds. It is an integer value.

International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 38 - 57, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1006

ArtNum: This feature refers to the article or product number associated with the transaction. It is an integer value.

TNcash: This feature is a boolean value that indicates whether the transaction was carried out using cash as the payment method (True) or not (False).

TNcard: This feature is a boolean value that indicates whether the transaction was carried out using a card as the payment method (True) or not (False).

Amount: This feature refers to the monetary value of the transaction in the local currency. It is a float value

OperatorId: This feature refers to the unique identifier of the operator who carried out the transaction. It is an integer value.

4.2 Feature Selection and Engineering

The features selected from the dataset in our experiment were: TranTime, BreakTime, ArtNum, Amount because the data points were clearly separable on the pairplot chart as shown in Figure 7.

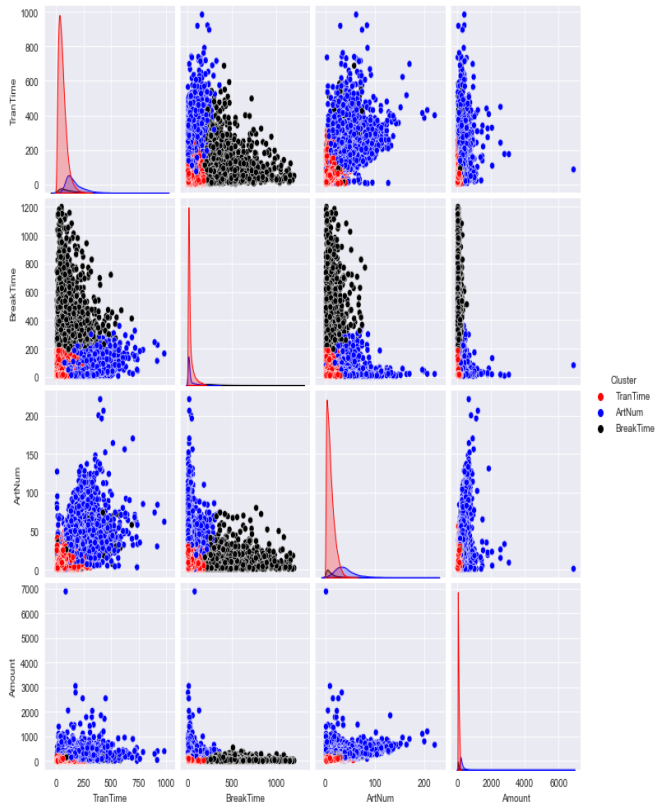

Figure 8 Pairplot Showing Separability of the Selected Features.

The pairplot in figure 8 also referred to as scatterplot matrix is used to understand and get insights into the best set of features to explain the relationship between two variables or to form the most separated clusters. It is particularly useful when there are multiple features and there is a need to understand how they relate with one another. It also aids in forming simple classification models by drawing some simple lines or make linear separation in the dataset. Diagonal Plots represent the distribution of a single variable uusually histograms or kernel density plots.Off-diagonal Plots are scatter plots showing the relationship between two variables. The colors represent different clusters. The diagonal plot indicates a Kernel Density Estimation (KDE) which is a way of looking at the distribution of the data. The pairplot creates a scatterplot which is colored according to its cluster assignment. This is used as a visualization tool for exploring how well-separated are the fratures, and based on the selected feature. If clusters are well-separated in some pairs of features but not in others, this can suggest that the well-separated features are more important for distinguishing between clusters. For example, if one variable tends to increase as the other increases, an upward-sloping trend in the scatter plot is visible.The pair plot also provides a comprehensive view of the pairwise relationships between all variables in the dataset, which can be particularly useful in multivariate analysis. In Figure 8, We have four features and are trying to create a pair of plots, we have four combination two ($4C_2 = 6$) which equates to 6 plots for visualization above and below the diagonal. The trantime (red color) feature in the x column and the breaktime (black color) feature are well separated in the second column of the grid implying that, this well separated features are important in our clusters. Also, the ArtNum(blue color) feature is well separated from the trantime feature in the third column and first row of the grid this further imply that a good model can be developed from these features. Likewise, the breaktime feature and artnum are well separated in the second column and third row of the grid which implies its suitability as a good feature to build a model. We can infer from the plot in Figure 8, in row 3 column 2 that as the ArtNum(the product number associated with the transaction) feature increases, the breaktime(idletime) feature increases forming a slightly gentle slope. This implies that there is a correlation between these two features.

In summary, the clusters are well separated especially between the red(trantime) and the blue (ArtNum). Additionally, there is a positive correlation trantime and breaktime.

4.3 Program Algorithm

The algorithm combines K-means clustering and K-nearest neighbors' classification to cluster the data and predict the cluster labels of new data points. The accuracy and other metrics help assess the performance of the clustering and classification tasks. Figure 8 shows the schematic representation of the program algorithm.

INPUT	OUTPUT
CSV file path containing POS transaction data.	Silhouette Score, Cluster Centers, Accuracy of KNN classifier, Classification Report, Confusion Matrix, and a

International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 38 - 57, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1006

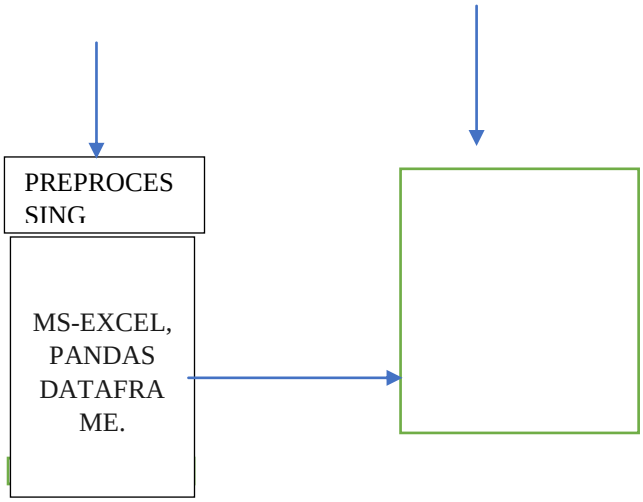

Figure 9 Schematic Diagram of the Program Algorithm

The following detailed processes were performed:

1. **Data Loading and Preprocessing:**
   - Import required libraries: pandas, numpy, KMeans from sklearn.cluster, train_test_split, silhouette_samples, silhouette_score, KNeighborsClassifier, accuracy_score, classification_report, confusion_matrix from sklearn.metrics, and matplotlib. pyplot and seaborn for data visualization.
   - Read the data from the CSV file 'dataset/POS_transactions_data.csv' into a pandas DataFrame named pos_df.
   - Select the numeric columns "TranTime", "BreakTime", "ArtNum", and "Amount" from the DataFrame and store them in X.
2. **Choose the Elbow Method for Optimal Cluster Choice:**
   i. Calculate KMeans clustering with different values of n_clusters (ranging from 2 to 10).
   ii. Compute the inertia (sum of squared distances from each data point to its assigned cluster center) for each n_clusters and store them in the list inertias.
   iii. Plot a graph to visualize the relationship between the number of clusters and the inertia to identify the optimal number of clusters (K) using the elbow method.
3. **Apply KMeans Clustering:**
   - Choose the number of clusters (K=3) based on the Elbow Method and create a KMeans model with n_clusters=3 and random_state=42.
   - Fit the model to the data points (X).
4. **Cluster Centers:**
   - Get the cluster centers' coordinates and create a DataFrame named cluster_centers_df with the columns "TranTime", "BreakTime", "ArtNum", and "Amount" to represent the centers of each cluster.
5. **Silhouette Score:**
   - Calculate the Silhouette clustering result with n_clusters=3 of the clusters.
6. **Assign Cluster Labels to Data Po**
   - Predict the cluster labels using the trained KMeans model.
   - Create a new DataFrame X_with_labels by concatenating the original X with an additional column "Cluster" containing the cluster labels.
   - Map the cluster labels to their corresponding feature names using the label_map dictionary.
7. **Data Visualization:**
   - Create a pairplot of the data points in X_with_labels, with points colored by their cluster assignment.
8. **Splitting Data into Train and Test Sets:**
   - Split the data into training and testing sets using train_test_split, with a test size of 20% and a random state of 42.
9. **Training KNN Classifier:**
   - Create a KNN classifier with n_neighbors=3 (3 nearest neighbors) and train it on the training data.
10. **Predicting Cluster Labels for the Test Set:**
    - Use the trained KNN model to predict the cluster labels for the test data.
11. **Evaluating KNN Model Accuracy:**
    - Compare the predicted cluster labels (y_pred_clusters) with the true cluster labels (y_test) and calculate the accuracy of the KNN model.
12. **Classification Report and Confusion Matrix:**
    - Generate a classification report to show precision, recall, F1-score, and support for each cluster label.

- Calculate the confusion matrix to evaluate the KNN model's performance.

13. Visualization of Confusion Matrix:

- Create a heatmap of the confusion matrix using seaborn to visualize the KNN model's performance.

Below is the detailed Algorithm for our program:

1. **Input**: CSV file path containing POS transaction data.
2. **Output**: Silhouette Score, Cluster Centers, Accuracy of KNN classifier, Classification Report, Confusion Matrix, and a heatmap.

Algorithm Steps:

Begin.

Step 1: Import the required libraries: (Numpy, sklearn, seaborn, pandas, matplotlib).

Step 2: Read data from the dataframe.(pos_df).

Step 3: Select Numeric Columns.

-TranTime, BreakTime, ArtNum

Step 4: Use the Elbow Method to determine the optimal number of clusters.

- Set a range of cluster options from 2 to 10.

-for each cluster option:

a. Apply KMeans with the given number of clusters to 'X'.

b. Calculate the inertia (within-cluster sum of squares) and store it in the 'inertias' list.

Step 5: Apply KMeans clustering with the selected number of clusters (3 in our case) to 'X'.

- for n_clusters in Option:
  - Apply KMeans clustering with given number of Clustering.
  - Calculate the inertia (within-cluster sum of square) and store it in the inertia list.
  - Plot the number of clusters (K) against the inertia values to find the "elbow point," which indicates the optimal number of clusters.

Step 6: Compute and display the cluster centers for each cluster.

Step 7: Predict cluster labels for the data points in 'X' using the trained KMeans model.

Step 8: Compute the Silhouette Score to evaluate the clustering performance.

Step 10: Create a mapping from cluster labels to feature names (‘numeric_cols’).

Step 11. Add the predicted cluster labels to 'X' DataFrame and map the cluster labels to feature names.

Step12. Visualize the data using a pairplot colored by cluster assignment.

Step13. Split the data into train and test sets (80-20 split).

Step 14. Train a K-nearest neighbors (KNN) classifier with the number of neighbors set to 3 using the training data.

Step 15. Predict cluster labels for the test set using the KNN classifier.

Step 16. Evaluate the accuracy of the KNN classifier by comparing the predicted cluster labels with the true cluster labels from the test set.

Step 17. Print the accuracy of the KNN classifier.

Step 18: Print a classification report, which includes precision, recall, and F1-score for each cluster label.

Step 20: Create a confusion matrix to visualize the classification performance of the KNN classifier.

Step 21: Generate a heatmap of the confusion matrix, where the x-axis and y-axis are labeled with the predicted and true cluster labels, respectively.

End.

4.4 Evaluation Metrics

To evaluate our proposed model, we employed the use of the following metrics:

1. **Accuracy:** Refers to the ratio of correctly predicted observation to the total observation.

In machine learning, accuracy is one of the metrics used to evaluate the performance of a classification model. It measures the proportion of correctly classified instances out of the total instances in the dataset. In other words, it tells us how many predictions made by the model were correct compared to the actual labels.

To understand accuracy in the context of a confusion matrix, let's first define what a confusion matrix is:

A confusion matrix is a table that is used to evaluate the performance of a classification model. It presents a summary of the model's predictions on a classification problem where the true values of the target variable are known. The matrix consists of four components:

1. **True Positives (TP):** The number of instances that are correctly predicted as positive by the model.
2. **True Negatives (TN):** The number of instances that are correctly predicted as negative by the model.
3. **False Positives (FP):** The number of instances that are incorrectly predicted as positive by the model (i.e., the model predicted positive, but the true label was negative).
4. **False Negatives (FN):** The number of instances that are incorrectly predicted as negative by the model (i.e., the model predicted negative, but the true label was positive).

Using these components, we can define accuracy as follows:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Accuracy represents the overall correctness of the model's predictions. It measures the percentage of correct predictions out of all the predictions made by the model. It is a common metric, especially when the class distribution is relatively balanced, meaning the number of instances in each class is similar.

However, accuracy may not always be the best metric to evaluate the model's performance, especially in cases where the class distribution is highly imbalanced. For instance, if the positive class is rare, and the model predicts most instances as negative, the accuracy might be high even though the model is not performing well on the positive class.

In such cases, it is essential to consider other metrics like precision, recall, F1-score, or area under the ROC curve (AUC-ROC) to get a more comprehensive view of the model's performance and how well it is performing for each class.

1. **Precision**: Precision, also referred to as positive predictive value, measures the proportion of true positive predictions (correctly predicted positive instances) out of all positive predictions made by the model. It is a measure of how many of the predicted positive instances are actually positive.

Precision= TP / (TP + FP).

High precision indicates that the model has a low rate of false positives, meaning that when it predicts a positive class, it is likely to be correct.

2. **Recall**: Recall, also known as sensitivity or true positive rate, measures the proportion of true positive predictions (correctly predicted positive instances) out of all actual positive instances in the dataset. It is a measure of how many of the actual positive instances are correctly predicted by the model. Usually given as:

Recall= TP / (TP + FN).

High recall indicates that the model has a low rate of false negatives, meaning that it correctly identifies most of the positive instances.

3. **F1-score**: The F1-score is the harmonic mean of precision and recall. It is used to balance the trade-off between precision and recall. The F1-score provides a single metric that takes both precision and recall into account. It is expressed as:

F1-score = 2 * (Precision * Recall) / (Precision + Recall).

The F1-score ranges from 0 to 1, where 1 represents a perfect model, and 0 indicates poor performance.

The F1-score is particularly useful when we want to find a balance between precision and recall. It is commonly used when the class distribution is imbalanced, and the model needs to perform well for both positive and negative classes.

5. RESULTS ANALYSIS AND DISCUSSION

5.1 Overview

In this chapter, we present the methodology and results of our research on an improved security for point of sale (POS) systems. The study aimed to explore the current security challenges faced by POS systems and to propose a new architecture that analyzes and identify these challenges.

Based on the findings of the literature review and case studies, we propose a new security architecture design for POS systems that identify the following key elements as possible solutions to the various attacks already discussed:

1. End-to-end encryption: To protect sensitive data, such as credit card numbers and personal information, from being intercepted or stolen during transmission.
2. Tokenization: To replace sensitive data with unique, non-sensitive tokens, reducing the risk of data breaches.
3. Multi-factor authentication: To provide an additional layer of security by requiring multiple forms of identification, such as a password and a fingerprint, to access the POS system.
4. Regular security assessments: To identify and address vulnerabilities in the system on a regular basis.

5.Employee training: To ensure that all employees understand the importance of security and know how to follow best practices to protect sensitive data.

The proposed design architecture was however not tested through a pilot implementation. It is the researcher’s conviction that the new architecture if implemented will be able to significantly improve the security of the retailer's POS system, reducing the risk of data breaches and other security incidents.

5.2 Exploratory Data Analysis

Table 2 An Overview Information of POS dataset using the Pandas Dataframe

<class 'pandas. core.frame.DataFrame'>

Range Index: 66863 entries, 0 to 66862

Data columns (total 11 columns):

#	Column	Non-Null Count	Dtype
0	WorkstationGroupID	66863 non-null	int64
1	TranID	66863 non-null	float64
2	BeginDateTime	66863 non-null	object
3	EndDateTime	66863 non-null	object
4	TranTime	66863 non-null	int64
5	BreakTime	66863 non-null	int64
6	ArtNum	66863 non-null	int64
7	TNcash	66863 non-null	bool
8	TNcard	66863 non-null	bool
9	Amount	66863 non-null	float64
10	OperatorId	66863 non-null	int64

dtypes: bool (2), float64(2), int64(5), object(2)

Memory usage: 4.7+ MB

The dataset consists of 66863 entries and 11 columns.

5.3 Result Analysis and Discussion

We employed KMeans clustering algorithm from the Scikit-learn library to group similar POS data points into clusters based on their similarity. Our results show the relationship between the number of clusters and the inertia value of a KMeans Clustering Algorithm. The inertia value is a measure of how internally coherent the clusters are.

There is a point in the plot where the decrease in inertia value begins to slow down. This point is known as the elbow point.

The elbow method was used to determine the optimal number of clusters for the data. This is shown in Figure 9.

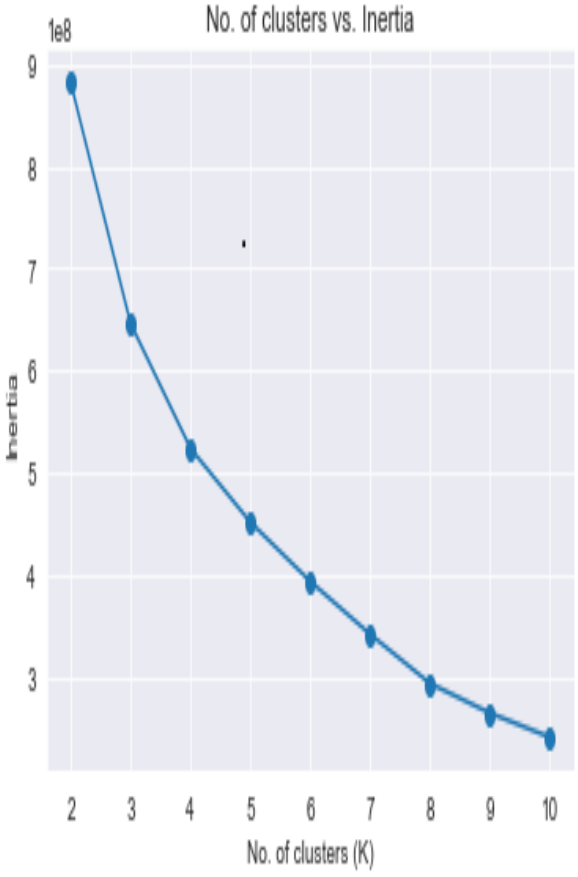

Figure 10The Elbow Method of Optimal Cluster Selection

The elbow represents a trade-off (balancing) between the number of clusters and the quality of the clustering. Choosing a larger number of clusters may result in better internal coherence within the clusters, but it may also lead to overfitting and reduced interpretability. On the other hand, choosing a smaller number of clusters may result in oversimplification and reduced accuracy.

By analyzing the elbow plot, we choose in our experiment three (3) as the optimal number of clusters for the dataset. Looking at Figure 10, at the third cluster, the line starts to deviate to the right forming the elbow. We choose the number of clusters at the elbow point, because it represents the best balance between the quality of the clustering and the complexity of the model.

In our experiment, we chose a random state parameter of 42 to ensure reproducibility of the results. (That is the number to get the same results when the experiment is to be conducted).

After selecting the number of clusters, we use KMeans clustering algorithm to apply to the dataset and the cluster centers for each cluster is printed. The cluster centers provide insight into the characteristics of the data points in each cluster and can be used for interpretation and prediction. After, we provide a convenient way to summarize and interpret the characteristics of each cluster in terms of

the mean value of each feature, and to associate each cluster label with its corresponding feature name. This is to help us predicts the cluster labels.

The silhouette score function from the Sklearn. metrics module is used to calculate the Silhouette score. It takes the data points X and the predicted cluster labels predictions as inputs and returns the Silhouette score as output. It shows how The Silhouette score is a measure of how well each data point fits into its assigned cluster compared to other clusters. Using the 3 as the number of clusters, the accuracy of Silhouette Score(n=3): was **0.581** percent.

The Neighbors Classifier is used to instantiate a KNN model with neighbors set to 3. This means that the model will consider the 3 nearest neighbors (clusters) of a data point when making a classification. The result bellow was obtained:

Table 3 Classification Report

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10835
1	0.97	0.99	0.98	534
2	0.99	0.98	0.99	2004
accuracy			1.00	13373
macro avg	0.99	0.99	0.99	13373
weighted avg	1.00	1.00	1.00	13373

The accuracy score represents the percentage of correctly classified instances in the test set, based on the KMeans cluster labels. In other words, it indicates how well the KNN model was able to classify new, unseen data points into the same clusters as the KMeans model. A higher accuracy score indicates better performance of the KNN model in classifying the test data. In our model, the accuracy score is used to calculate the accuracy of the predictions made by the KNN model on the test data (y_pred) compared to the actual labels of the test data (y_test).

This classification report provides an evaluation of the performance of a classifier for a multi-class classification problem with three classes, denoted by 0, 1, and 2. The report includes several metrics that measure the classifier's performance, such as precision, recall, and F1-score.

Precision: is the proportion of true positive predictions out of all the positive predictions made by the model. In other words, precision measures the accuracy of positive predictions. From Table 3, A precision of 1.0 means that all positive predictions made by the model are correct.

Recall: is the proportion of true positive predictions out of all the actual positive cases in the data. Recall measures how well the model identifies positive cases. A recall of 1.0 means that the model identifies all positive cases in the data.

F1-score: is the harmonic mean of precision and recall. It is a balanced measure that combines precision and recall into a single score. The F1-score is useful when the data is imbalanced, i.e., one class has many more samples than the other classes.

Support: is the number of samples in each class.

The report also includes an accuracy score, which measures the proportion of correctly classified samples out of all the samples in the dataset.

The macro-averaged metrics take the average of the metrics computed for each class, giving equal weight to each class. The weighted-average metrics, on the other hand, take the average of the metrics weighted by the number of samples in each class, giving more weight to classes with more samples.

In this particular report, the classifier achieved very high performance, with an accuracy of 1.0, indicating that all samples in the dataset were classified correctly. The precision for class 0 is also 1.0, indicating that all samples that were predicted as class 0 were correct. The precision for class 1 and class 2 are also high, at 0.97 and 0.99 respectively. The recall values are also high, with all classes having a recall of at least 0.98, indicating that the classifier identified almost all the positive cases in the dataset. The F1-scores are also high, indicating a good balance between precision and recall. In general, the classifier has performed very well on this particular dataset, with high precision, recall, and F1-scores across all classes. The result of KNN classifier can also be visualized in the confusion matrix as shown in Figure 11, The heatmap helps visualize the performance of the classifier by displaying the number of correctly and incorrectly classified instances in each class.

International Journal of Computer Applications Technology and Research
Volume 13–Issue 02, 38 - 57, 2024, ISSN:-2319–8656
DOI:10.7753/IJCATR1302.1006

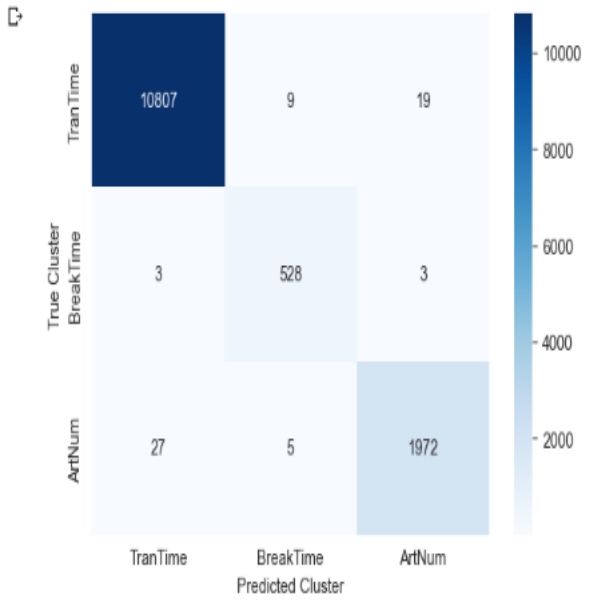

Figure 11 Confusion Matrix K-Means Clustering

The diagonal cells of the heatmap in Figure 11 represent the number of observations that were correctly classified. These cells show the number of test data points that were assigned to their true cluster label. The TranTime feature has 10807 data points that were correctly predicted. Also, from the confusion matrix, the BreakTime feature has 528 correctly predicted data points and the Artnum feature has 1972 correctly predicted points respectively. The non-diagonal cells represent the number of observations that were assigned to the wrong cluster or misclassified. The rows of the heatmap represent the true cluster labels, and the columns represent the predicted cluster labels. The numbers outside the diagonal of the confusion matrix represent the misclassifications, i.e. the instances that were predicted to belong to a different cluster than their actual cluster. These values give us an idea of which clusters are being confused with which other clusters.

For example, if we look at the row for Cluster 0 in the confusion matrix and see that there are some non-zero values in the columns for Clusters 1 and 2, it means that some of the instances that actually belong to Cluster 0 were misclassified as belonging to Clusters 1 or 2. Similarly, if we look at the column for Cluster 1 and see that there are some non-zero values in the rows for Clusters 0 and 2, it means that some of the instances that were predicted to belong to Cluster 1 actually belong to Clusters 0 or 2.

Overall, the confusion matrix gives us a more detailed view of the performance of the KNN classifier using the KMeans cluster labels as compared to just looking at the accuracy score. It allows us to see which clusters are being confused with each other and can help us identify patterns in the misclassifications that might inform future improvements to the clustering or classification models.

6. CONCLUSION

This research was aimed at identifying the key security breaches faced by POS systems and proposed a new architecture that addresses these challenges and proposes the use of end-to-end encryption, tokenization, multi-factor authentication, regular security assessments, and employee training. This research also used three clusters to separate POS data in order to mine similar patterns in the dataset. From the result obtained in the experiment, we can conclude that using KMeans clustering and KNearest Neighbour (KNN) algorithm presents the best algorithms to detect intrinsic patterns and it can be used to detect fraudulent transaction patterns in POS transactions.

6.1 Limitation of the work

Our experiment assumes our dataset contains fraudulent transactions or datapoints. The proposed architecture has not been tested practically.

6.2Recommendation

The improved architecture developed to identify and mitigate against point-of-sale breaches if adopted can be addressed the problem of point-of-sale system breaches and improve fraud detection. However, further study should be carried out in the combination of other clustering techniques (DBScan or LSTM) to identify and detect intrinsic patterns in POS dataset and the use of machine learning techniques to identify and detect fraud in keystroke dynamics of point-of-sale interface.

6.3 Contribution to knowledge

This research provides a novel architectural design that identifies areas or points of likely breach or attack in a point-of-sale system. Again, it performs a classification task based on the POS features and achieved a high accuracy of 99.51% which is very important to fraud detection in point-of-sale systems.

7. ACKWOLEDGEMENTS

My Sincere gratitude goes to Almighty God for the gift of knowledge and wisdom and for making this research possible.

To Dr. Aamo Iorliam, thank you so much for your guidance and direction all through this research.

I will also want to appreciate, Dr. Otache Innocent Ogwuche, Dr. Patrick Obilikwu, Dr. Adeyelu, A. A, Mr. Tivlumun Ge for their contributions. My appreciation also goes to my parents Capt. M. T. Anji (Rtd) and Mrs. Elizabeth M. Terwase for their love and support. My profound gratitude goes to my wife, Mrs. Terwase Agnes Edugwu, my children: Sophia, Fortune, Michael (Jnr), my siblings,

course mates and friends. Thank you for all your support, May God bless you all.

8. REFERENCES

[1] Abell, J. C. (2009, November 4). Nov. 4, 1879: Ka-Ching! The World's First Cash Register. Retrieved from Nov. 4, 1879: Ka-Ching! The World's First Cash Register | WIRED.

[2]Akinyele, A. I., Muturi, W., & Ngumi, P. (2017). Financial innovation and fraud risks in deposit money banks of Nigeria. EPRA *International Journal of Economic and Business Review*, 3(12), 56-66.

[3] Adegboyega, J. E., & Tomola, M. O. (2018). Card Frauds and Customers' Confidence in Alternative Banking Channels in Nigeria.DOI:10.19044/esj.2018.v14n16p40. URL:http://dx.doi.org/10.19044/esj.2018.v14n16p40.

[4] Alabi, O. F., & David, A. (2022). Framework For Detection Of Fraud At Point Of Sale On Electronic Commerce Sites Using Logistic Regression. https://doi.org/10.21203/rs.3.rs-1699624/v1

[5] Amaefule, I. A., & Njoku, D. O. (2019). The Prospect and Challenges of POS as Electronic Payment System in Nigeria. International Journal of Scientific Research and Management.

[6] Bandhakavi, S., Bisht, P., Parthasarathy, M., & Venkatakrishnan, V. (2007). CANDID: Preventing SQL injection attacks using dynamic candidate evaluations. *Journal Name, Volume Number(Issue Number)*, 12-24.https://doi.org/10.1145/1315245.1315249

[7] Bernardi, M. L., Cimitile, M., Di Francescomarino, C., & Maggi, F. M. (2014). Using discriminative rule mining to discover declarative process models with non-atomic activities.

[8] Coates, A., & Ng, A. Y. (2012). Learning Feature Representations with k-means. In *Advances in Neural Information Processing Systems* (pp. 561–580).

[9] Davis, W. C., & Wang, Z. J. (2015). A mobile retail POS: Design and implementation. In *Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia* (pp. 426-432). ACM. http://dx.doi.org/10.1145/2757384.2757391.

[10] Dey, A., Jain, S., & Nandi, S. (2019). New method of pos based on artificial intelligence and cloud computing (pp. 1-6). https://doi.org/10.1109/ICRAECC43874.2019.8995078.

[11] Drimer, S., Murdoch, S. J., & Anderson, R. (2014). Security failures in smart card systems: tampering the tamper-proof. http://www.cl.cam.ac.uk/users/{sd410, sjm217, rja14} a. EE/library/WP_M-Trends2014_140409.pdf.

[12] Feit, A. M., Weir, D., & Oulasvirta, A. (2016). How we type: movement strategies and performance in everyday typing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 4262-4273). ACM. https://doi.org/10.1145/2858036.2858233

[13] Goldman, J. (2021). Point of sale security measures for 2022.

[14]Gomzin, S. (2014). Hacking Point of Sale: Payment Application Secrets and Threats.

[15] Gundert, L. (2014). Detecting payment card data breaches today to avoid becoming tomorrow's headline. Retrieved from http://blogs.cisco.com/security/

[16] Hines, C., & Youssef, A. (2018). Machine Learning Applied to Point-of-Sale Fraud Detection. In *Information Security and Privacy* (pp. 201–210). Springer. https://doi.org/10.1007/978-3-319-96136-1_23

[17] Kazi, M. S. (2013). Secure Mobile POS System (Master's Thesis). KTH Royal Institute of Technology, Stockholm, Sweden.

[18] Kaspersky Lab. (2017). Type of malware. Retrieved from http://usa.kaspersky.com/internet security-center/threats/malware-classifications#.U755xLEe8SR

[19] Kundai, S. (2017). An analysis of point of sale systems physical configurations and security measures in Zimbabwean SMEs. *IRA International Journal of Education and Multidisciplinary Studies, 6*(2), 5.

[20] Lord, N. (2021). Data protection 101. What is POS. Protecting Data in POS Environment.

[21] Mandiant, R. (2014). M-Trends: Beyond the breach. Retrieved from https://dl.mandiant.com.

[22] Nigerian Inter-bank Settlement System.

[23] Nigeria Electronic Fraud Forum Annual Report 2016).

[24] The Register@-Biting the hands that feeds IT. Mon, 24 Oct; 2022 //22:11 UTC.
The Unicode Consortium. (2011). The Unicode Standard. Available at https://unicode.org/standard/standard.html.

[25] Trend Micro Inc. (2014). Point-of-sale system breaches: Threats to the retail and hospitality industries.Retrievedfromhttp://www.trendmicro.com/cloud content/us/pdfs/securityintelligence/white-papers/wp-pos-system-breaches.pdf.

[26] Thomas, A., & Rafal, W. (2019). Point of sale (POS) data from a supermarket: Transactions and cashier operations.

[27] Whitteker, W. (2014). Point of sale system and security. *GIAC Gold Certification*. Wired. Retrieved from http://www.wired.com/2009/11/1104ritty-cash-register.

[28] Zhao, X., Chen, S., Zhou, L. and Chen, Y. (2020). Sound source localization based on SRP-PHAT spatial spectrum and deep neural network. *Computers, Materials & Continua64*(1), 253–271 DOI 10.32604/cmc.2020.09848.