IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, November 2024

Intrusion Detection System Using Machine Learning

Mr. Mahendra Sanjay Dalvi[1] and Dr. Nilesh R. Wankhade[2]
Student, Department of Computer Engineering[1]
Head of Department, Department of Computer Engineering[2]
Late G. N. Sapkal College of Engineering, Nashik, India

Abstract: *The system administrator can find network security breaches in their own organization with the aid of a system network intrusion detection system (NIDS). However, several challenges arise when developing a sophisticated and potent NIDS for unforeseen and erratic attacks. One of the main areas of interest in NIDS research in recent years has been the use of machine learning techniques. This system proposes a network intrusion detection technique that effectively detects various types of network intrusion i.e., Dos, U2R, R2L, Probe, Normal. It is based on decision tree and twin support vector machine.The decision tree for the network traffic data is first constructed using the trees. The bottom-up merging approach is then used to maximize the separation of the decision tree's upper nodes, thereby reducing the accumulation of errors during the decision tree's construction. Subsequently, the network intrusion detection model is implemented by embedding twin support vector machines into the decision tree. This performance evaluated network intrusion detection analysis dataset, particularly KDD-CUP99,NSLKDD dataset.*

Keywords: Network intrusion detection, Decision Tree, win support vector machine, Encoder, KDD Dataset

I. INTRODUCTION

The primary challenge with current Network Intrusion Detection Systems (NIDS) lies in their reliance on signature-based techniques, which struggle to detect novel or evolving threats, and their inability to process the vast and diverse nature of modern network data effectively. Traditional methods are further hampered by limited monitoring granularity and the high volume of traffic, which reduces the accuracy and speed of threat detection. As cyber-attacks grow more sophisticated, techniques such as Denial of Service (DoS) attacks and advanced malware can bypass these systems, exposing critical vulnerabilities. While machine learning has shown potential in enhancing NIDS, shallow learning approaches often fail to generalize well across different data environments. To address these limitations, a deep learning-based NIDS offers promise by automatically learning complex features from raw traffic, enabling more accurate and identification of known and new threats in real-time across a variety of protocols and data formats, while also reducing false positives and improving network security resilience.

II. OBJECTIVE & SCOPE OF PROPOSED SYSTEM

Current methods' performance and accuracy can be enhanced to develop a method that can deliver dependable supervised feature learning.

- To research the various kinds of Network Intrusion Detection Systems (NIDSs) currently in use.
- To research different machine learning techniques for classifying traffic.
- To investigate stacking non-symmetric deep auto-encoders for the extraction of unsupervised features.
- To examine the outcomes of experiments using the suggested Twin Support Vector Machine and Decision Tree classification algorithms for intrusion detection systems.
- Decrease the amount of time spent training.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, November 2024

III. FEATURES OF PROJECT

- Real-time monitoring
- Fraud detection
- Anomaly-based Detection
- Real-time Alerts and Notifications
- Logging and Reporting
- Network performance metrics
- Multi-Platform Support

IV. LITERATURE REVIEW

B. Dong and X. Wang concentrate on deep learning techniques that learn from lower-level traits to higher-level concepts, drawing inspiration from the layered organization of the human brain. The Deep Belief Network (DBN) can translate functions from input to output without the need for human-crafted features thanks to this multi-level abstraction. The Restricted Boltzmann Machine (RBM) is an unsupervised learning technique used in each layer of a DBN. One of DBN's benefits is its deep coding capacity, which enables it to conduct thorough data analysis and adjust to shifting data settings. DBN is also useful for anomaly detection, which includes spotting odd traffic patterns and system behavior. But the method necessitates quicker and more effective data processing [2].

Research on the use of deep learning for machine health monitoring is reviewed and summarized by R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao. With an emphasis on Autoencoders (AEs) and their variations, Restricted Boltzmann Machines (RBMs) and their variations (including Deep Belief Networks (DBNs) and Deep Boltzmann Machines (DBMs)), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs), the paper explores deep learning applications in machine health monitoring systems. DL-based Machine Health Monitoring Systems (MHMS) have the advantage of being more adaptable to different kinds of machines and requiring less human effort and specialized knowledge. However, a major drawback is that the size and caliber of the datasets employed have a big impact on how well DL-based MHMS performs [3].

H. Lee and Y. Kim suggest developing an FDC model that carries out concurrent feature extraction and classification by utilising a deep learning method called stacking denoisingautoencoder (SdA). The SdA model is resistant to measurement noise and can detect global and invariant features in sensor signals for fault monitoring. The model can learn global features from complicated input data, including multivariate time-series datasets and high-resolution pictures, thanks to a SdA, which is made up of denoisingautoencoders stacked layer by layer.Because the SdA model can learn both normal and fault-related features from sensor data without preprocessing, it has several advantages in real-world applications. One drawback, though, is that additional research into the trained SdA is required to determine which process factors have the biggest effects on classification outcomes [4].

For the automatic security audit of brief messages from prisons, L. You, Y. Li, Y and Y. Yang suggests a unique DL-based recurrent neural network (RNN) model that can distinguish between secure and unsecure messages. This work uses word2vec to map each sentence to a feature vector and extract features from brief communications by capturing word order information. RNNs categorise these vectors after words with comparable meanings are arranged similarly in the vector space. With an average accuracy of 92.7%, the RNN model outperforms the SVM. One benefit is the ability to combine several feature extraction and classification methods using ensemble frameworks, which could improve performance even more. One drawback is that the approach is only appropriate for brief messages and might not work well for lengthy ones [5].

Using a deep convolutional neural network on a cloud platform, R. Polishetty, M. Roopaei, and P. Rad provide a signature-based feature method for segmentation, character detection, and license plate localisation. In difficult situations, such as (i) crowded traffic with multiple plates in an image, (ii) varying plate orientation relative to brightness, (iii) additional information displayed on the plate, (iv) distortion from wear and tear, and (v) image distortion from bad weather conditions, like haze, the Plate Recognition System (LPRS) can accurately identify license plates by extracting important features. When compared to conventional LPRS techniques, the suggested algorithm

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, November 2024

offers the advantage of higher license plate recognition accuracy. The incidence of unidentified or incorrectly detected images is a disadvantage, even [6].

K. Alrawashdeh and C. Purdy provide a deep learning method for anomaly detection that makes use of a Deep Belief Network (DBN) and a Restricted Boltzmann Machine (RBM). This technique uses an unsupervised feature reduction single-hidden-layer RBM. A DBN is created by passing the weights that are produced by one RBM to another RBM. A layer employing a Logistic Regression (LR) classifier with multi-class softmax is used to further refine the pre-trained weights.

This method has the advantage of a low false-negative rate of 2.47% and an accuracy of 97.9%. Nevertheless, the approach might be strengthened by refining the dataset and the deep learning network's feature reduction procedure [7].

A. Javaid, Q. Niyaz, W. Sun, and M. Alam suggest a deep learning-based strategy for creating a Network Intrusion Detection System (NIDS) that is effective and adaptable. Using the NSL-KDD benchmark dataset for network intrusion, the suggested NIDS applies Self-taught Learning (STL), a deep learning technique, using a sparse autoencoder and softmax regression. One benefit of this strategy is that STL achieved a classification accuracy rate of more than 98% for all classification classes. One drawback, though, is that this deep learning method requires the implementation of a real-time NIDS for real networks [8].

S. Potluri and C. Diedrich investigate how well an intrusion detection system (IDS) based on a deep neural network (DNN) handles massive amounts of network data using multi-core CPUs and GPUs. The neural network's parallel processing capabilities allow the DNN to process network traffic quickly and effectively. Reliability and efficiency in intrusion detection, especially in recognising particular attack classes with the required amount of training samples, are benefits of the DNN-based IDS. It was discovered that the multi-core CPU outperformed serial training methods. The requirement to increase detection accuracies for the DNN-based IDS is a disadvantage, though [9].

Replicator neural networks (RNNs) are used by C. Garcia Cordero, S. Hauke, M. Muhlhauser, and M. Fischer to develop anomaly detection models as part of their method for identifying extensive network-wide attacks. Accurately identifying network-wide anomalies without presuming that the training data is totally free of attacks, the method is unsupervised and does not require labelled data. The ability of the suggested methodology to successfully detect all well-known Distributed Denial of Service (DDoS) assaults and SYN port scans is one of its advantages. Furthermore, it is resistant to learning when attacked, which is a drawback in related efforts. One drawback is that stacked autoencoder deep learning techniques must be used to enhance the process [10].

T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho suggest a deep learning-based anomaly detection system that takes advantage of Software-Defined Networking's (SDN) flow-based architecture. Their method uses a deep learning model to detect anomalies in an SDN environment based on flow. Finding the Deep Neural Network's (DNN) ideal hyperparameters and verifying detection and false alarm rates are among the benefits.

With just six fundamental network features, the model's performance accuracy of 75.75% is fairly reasonable. One drawback is that the model might not function well in an actual SDN setting [11].

In a technique put forth by NileshWankhade, patterns connected to malicious activity are found in network logs or transaction datasets for intrusion detection systems (IDS) by applying data mining techniques like frequent itemset mining (specifically using the Apriori algorithm) and association rule mining. Lower-frequency patterns are filtered out as benign using the Apriori algorithm, which concentrates on high-frequency patterns that are statistically more likely to constitute invasions. Furthermore, transaction data is encrypted using the DES (Data Encryption Standard) method before to transmission from client to server, improving security by guaranteeing that only authorised parties are able to decrypt and examine the data. By securing communication between network nodes and the central monitoring system, this encryption lowers the possibility of unwanted access in the context of intrusion detection systems.[12]

V. REPRESENTATION OF THE METHODOLOGY

Figure shows the suggested intrusion detection system's framework. There are four primary stages to the detection framework:

(1) Data collection, which involves gathering network packet sequences.

(2) Data pre-processing, in which key characteristics that can set one class apart from the others are chosen after training and test data have been pre-processed.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, November 2024

(3) Classifier training, in which Twin SVM and Decision Tree (DT) are used to train the classification model.
(4) Attack recognition, in which intrusions on the test data are found using the learned classifier.

1. Data Collection: The first and most important phase in intrusion detection is data collection. Two crucial elements in the design and efficacy of an intrusion detection system are the kind of data source and the location from which data is gathered. This study suggests a network-based intrusion detection system (IDS) to verify our suggested methods and offer the most appropriate security for the targeted host or networks. The suggested intrusion detection system (IDS) examines incoming network traffic while operating on the router closest to the victim or victims. The gathered data samples are labelled against the domain knowledge and grouped according to the transport/Internet layer protocols during the training phase. But only the protocol types are used to categorize the data gathered during the test phase.

2. Data Pre-processing: The data obtained during the phase of data collection are first processed to generate the basic features such as the ones in KDD Cup 99 dataset. This phase contains three main stages shown as follows: Data Transferring, Data Normalization, Feature Selection.

3. Classifier Training: Following feature selection, the best subset of features is used in the classifier training step, when DT and Twin SVM are used. Classifiers must be used because DT and Twin SVM are limited to handling binary classification problems, and five ideal feature subsets are chosen for each class in the KDD Dataset. One class of records is distinguished from the others by each classifier. The Normal classifier, for instance, separates Normal data from non-Normal (all kinds of attacks). DoS traffic and non-DoS data, such as Normal, Probe, R2L, and U2R instances, are separated by the DoS class. The intrusion detection model is then constructed by combining the classifiers to differentiate between all classes.

4. Attack Recognition: The most correlated and significant features are included in the optimal subset of features used to train the classifier; the stored trained classifier can be used to distinguish between normal and intrusion data. The saved trained model is then used to identify intrusions in the test data. Attacks are recorded for records that do not meet the regular class, which are regarded as normal data. The subclass of the abnormal record (the kind of attacks) can be utilized to identify the type of record if the classifier model validates that the record is abnormal. output as normal or abnormal (accuracy of detection, rate of false positives, and time spent on detector development).

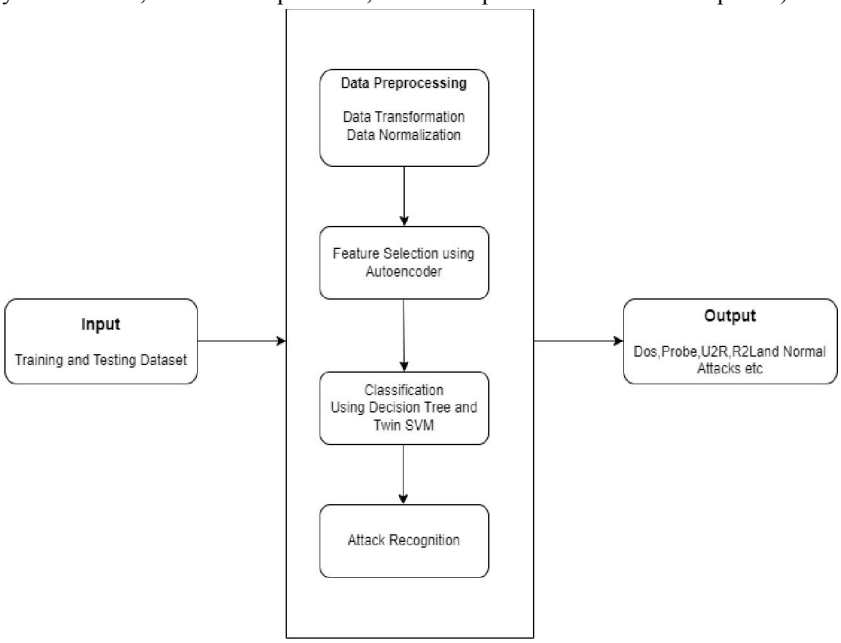

Fig :System Architecture

VI. ADVANTAGES

- Early Threat Detection: Quicker reactions to reduce damage are made possible by early detection of possible security breaches.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, November 2024

- Real-time Monitoring: Continuously monitors network traffic and system activities, helping to catch threats as they happen.
- Automated Responses: To lessen the impact of an attack, many IDS are able to act instantly, for example, by isolating compromised computers or blocking hostile IPs.
- Decreased Impact of Cyberattacks: Security teams can react swiftly thanks to alerts, which help to avoid or lessen data loss and harm.
- Aids in Compliance: By keeping logs, conducting audits, and monitoring security occurrences, it helps comply with regulatory standards.
- Minimizes False Alarms: Security teams can concentrate on real threats when advanced IDS, particularly ones with machine learning, minimize the number of false positives.
- Integration with Other Security technologies: Improves the overall security infrastructure by working well with firewalls, SIEMs, and other technologies.

By identifying any bottlenecks or inefficiencies, enhanced network performance analysis provides information that network administrators can use to optimize network performance.

VII. APPLICATION AREAS

- Network Security Monitoring: IDS is frequently used by businesses to defend internal networks against outside threats by continually monitoring network traffic to identify malicious activity.
- Critical Infrastructure Protection: Used to protect vital infrastructure from cyberattacks that could interrupt services or jeopardize public safety in industries like energy, utilities, and transportation.
- Banking and Financial Services: Prevents fraud, insider threats, and external assaults on ATMs, payment systems, and financial networks in order to safeguard sensitive financial data and client information.
- Government and Defense: Employed by government organizations, particularly those in the intelligence, defense,and vital services industries, to safeguard private data and fend against nation-state cyberattacks.
- Educational Institutions: Prevents cyberattacks on academic networks, protecting professor and student information and research data.

VIII. HARDWARE REQUIREMENTS

- CPU Intel / AMD 3.1Ghz (Intel i3 or better)
- Memory 4 GB
- The ability to install more memory is desirable. Disk 120 GB SSD or better
- Graphics Accelerated, Gaming Support Nvidia is preferred over AMD 1920 by 1080 resolution is recommended (at least on an external port) At least 1280 by 1024 resolution
- HDMI output recommended (perhaps with an adapter)
- Mouse An external mouse (USB or Bluetooth) is desirable.
- USB 3.0 desirable for an external disk Other USB port may be needed for: mouse, printer, mic-in, and headphones-out, depending on how these are connected.
- External monitor A 23” or larger HDMI monitor is recommended, with reasonable resolution.
- Laptop or Desktop Windows 11 or macOS 12.4 or above. Linux is also acceptable if a mainstream distribution (e.g., Ubuntu).

IX. SOFTWARE REQUIREMENTS

- Operating System: Windows7 and above versions
- Front End:HTML, CSS
- Programming Language: Java, Java Script
- Backend: MySQL
- Ide: Eclipse

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, November 2024

- Server: Apache Tomcat 7 and above
- Dataset: NSLKDD, KDD(Kaggle.com)
- Domain: Machine Learning
- Algorithm: TSVM, Decision Tree.

X. TEST DATA REQUIREMENTS

Unit Testing

Unit testing focuses on verifying the program's smallest component, the module. To identify faults within the module's boundaries, key control pathways are evaluated using the comprehensive design description. According to unit testing, every submodule in this system such as campaigns, leads, contacts, etc.is tested separately. They test their input field validations.

Integration testing

Following the testing of each individual unit, it is necessary to test the way the units were assembled to make sure that no data is lost across interfaces, that one module does not negatively affect another, and that a function is carried out appropriately. Following unit testing, each submodule is tested in relation to the others.

XI. SYSTEM TESTING FOR THE CURRENT SYSTEM

After integrating all of the project's primary components, we test the system as a whole at this testing stage. We are checking to see if the system is producing accurate results. Every module was integrated, and the information flow between them was examined. Additionally, the data flow was examined to see if it met the requirements. Additionally, it was examined whether any specific module was not functional, that is, whether or not all of the modules were functioning fully when the integration was complete.

Functional testing entails examining the system's operation to make sure it satisfies the necessary requirements and operates as planned. This entails evaluating the accuracy of the churn forecast, input data processing, and output interpretation.

- Performance testing: Testing the system's response time, resource usage, and scalability is part of this.
- Security testing: Testing the system's security features to make sure it can shield private client information from theft, misuse, and illegal access is known as security testing. Testing the system's authorization, encryption, and authentication processes is part of this.
- Compatibility testing: To make that the system can function in a range of situations, this entails testing its compatibility with various hardware configurations, databases, and operating systems.
- Usability testing: This entails evaluating the system's user interface and user experience to make sure it is user-friendly, intuitive, and satisfies end users' needs.
- Regression testing: This entails verifying the system's operation following upgrades or modifications to make sure there haven't been any unanticipated side effects or regressions.
- Acceptance testing: This entails evaluating the system's performance from the viewpoint of the end users to make sure it satisfies their needs and expectations. This involves evaluating the system's precision, dependability, and usability.
- Recovery testing: This entails evaluating the system's resilience to malfunctions, mistakes, or calamities to make sure it can keep running and serving end users.
- Stress testing: This entails evaluating the system's functionality under high load scenarios to make sure it can manage unforeseen or disastrous situations.
- Exploratory testing: This entails evaluating the system's performance and behavior under unforeseen or unexpected circumstances to make sure it can manage unforeseen circumstances and deliver precise and trustworthy results.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, November 2024

At this testing stage, we examined the following:

- Whether all the forms are properly working or not.
- Whether all the forms are properly linked or not.
- Whether all the images are properly displayed or not.
- Whether data retrieval is proper or not

XII. CONCLUSION

In conclusion, we've talked about the issues with current NIDS methods. We have developed our unique NDAE approach for unsupervised feature learning in response to this. We have since expanded on this by putting forth a brand-new classification model made using stacked NDAEs, the TSVM, and the DT classification method. We also put the intrusion protection system into place. The outcome demonstrates that our method provides excellent levels of recall, accuracy, and precision together with a shorter training period.

REFERENCES

[1]. LI ZOU, XUEMEI LUO, YAN ZHANG, XIAO YANG AND XIANGWEN WANG" HC-DTTSVM: A Network Intrusion Detection Method Based on Decision Tree Twin Support Vector Machine and Hierarchical Clustering"

[2]. B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in Proc. 8th IEEE Int.Conf. Commun. Softw. Netw, Beijing, China, Jun. 2016, pp. 581–585.

[3]. R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring: A survey," Submitted to IEEE Trans. Neural Netw. Learn. Syst., 2016. [Online]. Available:http://arxiv.org/abs/1612.07640

[4]. H. Lee, Y. Kim, and C. O. Kim, "A deep learning model for robust wafer fault monitoring with sensor measurement noise," IEEE Trans. Semicond. Manuf., vol. 30, no. 1, pp. 23–31, Feb. 2017

[5]. L. You, Y. Li, Y. Wang, J. Zhang, and Y. Yang, "A deep learning based RNNs model for automatic security audit of short messages," in Proc. 16th Int. Symp. Commun. Inf. Technol., Qingdao, China, Sep. 2016, pp. 225–229.

[6]. Polishetty, M. Roopaei, and P. Rad, "A next-generation secure cloud based deep learning license plate recognition for smart cities," in Proc. 15th IEEE Int. Conf.Mach. Learn. Appl., Anaheim, CA, USA, Dec. 2016, pp. 286–293.

[7]. K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in Proc. 15th IEEE Int. Conf. Mach. Learn. Appl., Anaheim, CA, USA, Dec. 2016, pp. 195–200.

[8]. A Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in Proc. 9th EAI Int.Conf. Bio-Inspired Inf. Commun. Technol., 2016, pp. 21–26. [Online]. Available: http://dx.doi.org/10.4108/eai.3-12-2015.2262516

[9]. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced intrusion detection system," in Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom., Berlin, Germany, Sep. 2016, pp. 1–8.

[10]. C. Garcia Cordero, S. Hauke, M. Muhlhauser, and M. Fischer, "Analyzing flow-based anomaly intrusion detection using replicator neural networks," in Proc. 14th Annu. Conf. Privacy, Security. Trust, Auckland, New Zeland, Dec. 2016, pp. 317–324.

[11]. T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho,"Deep learning approach for network intrusion detection in software defined networking," in Proc. Int. Conf. Wireless Netw. Mobile Commun., Oct. 2016,pp. 258–263.

[12]. Chopda, K., Rote, A., Gaikwad, K., Gachale, P., &Wankhade, N. R. (2017). Association Rule Mining Method for Applying Encryption Techniques in Transaction Data. International Journal of Engineering Science and Computing