IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A WELL INFORMED EXPOSURE WITH ONE CLASS SUPPORT TUCKER MACHINE AND GENETIC ALGORITHM TOWARDS BIG SENSOR DATA IN IOT

P.PRIYADHARSHINI[1], ULAGALAKSHMI K [2], VIGNESHWARI R[3,]

1 ASSISTANT PROFESSOR, 2,3 UG STUDENTS
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAHENDRA ENGINEERING COLLEGE, TAMILNADU, INDIA

ABSTRACT

Various types of sensor data can be collected by the Internet of Things (IoT). Each sensor node has spatial attributes and may also be associated with a large number of measurement data that evolve over time; therefore, these highdimensional sensor data are inherently large scale. Detecting outliers in large-scale IoT sensor data is a challenging task. Most existing anomaly detection methods are based on a vector representation. However, large-scale IoT sensor data have characteristics that make tensor methods more efficient for extracting information. The vector-based methods can destroy original structural information and correlation within large-scale sensor data, resulting in the problem of the "curse of dimensionality," and some outliers hence cannot be detected. In this paper, we propose a one-class support Tucker machine (OCSTuM) and an OCSTuM based on tensor Tucker factorization and a genetic algorithm called GA-OCSTuM. These methods extend one-class support vector machines to tensor space. OCSTuM and GA-OCSTuM are unsupervised anomaly detection approaches for big sensor data. They retain the structural information of data while improving the accuracy and efficiency of anomaly detection. The experimental evaluations on real datasets demonstrate that our proposed method improves the accuracy and efficiency of anomaly detection while retaining the intrinsic structure of big sensor data.

Keywords : Tucker machine, sensor data & IOT sensor

1.INTRODUCTION

The Internet of Things (IOT) is made up of devices that have the capability to sense, communicate, compute, and even control their environment. These devices are increasingly becoming a part of complex, dynamic, and distributed networks such as electricity or mobile networks. A variety of sensors in IoT can capture multiple aspects of the environment that they

monitor in real time. For instance, phasor measurement sensors capture transient dynamics and evolving disturbances in the power system in a synchronized manner in real time. In traffic networks, a car currently can deliver about 250 GB of data per hour from its IoT-connected electronics such as weather sensors, parking cameras, and radar. The IoT will consist of about 50 billion objects by 2020, which will trigger the era of large-scale computing, necessitating the management of computing heat and energy in concert with increasingly more complex processor/network/memory hierarchies of sensors and embedded computers in distributed systems. The data representation of the raw data from such systems is crucial for the extraction of relevant information. It is very important that information extraction in IoT uses efficient

1.2 BIG DATA - OVERVIEW

Big data is an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process them using traditional data processing applications. The challenges include analysis, capture, duration, search, sharing, storage, transfer, visualization, and privacy violations. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, prevent diseases, combat crime and so on."

Scientists regularly encounter limitations due to large data sets in many areas, including meteorology, genomics, connectomics, complex physics simulations, and biological and environmental research. The limitations also affect Internet search, finance and business informatics. Data sets grow in size in part because they are increasingly being gathered by ubiquitous informationsensing mobile devices, aerial sensory technologies (remote sensing), software logs, cameras, microphones, radio-frequency identification (RFID) readers, and wireless sensor networks. The world's technological per-capita capacity to store information has roughly doubled every 40 months since the 1980s; as of 2012, every day 2.5exabytes (2.5×10^{18}) of data were created. The challenge for large enterprises is determining who should own big data initiatives that straddle the entire organization.

1.3 Descriptive and Statistical Information of Big Data

If Gartner’s definition (the 3Vs) is still widely used, the growing maturity of the concept fosters a more sound difference between big data and Business Intelligence, regarding data and their use:

- Business Intelligence uses descriptive statistics with data with high information density to measure things, detect trends etc.;
- Big data uses inductive statistics and concepts from nonlinear system identification to infer laws (regressions, nonlinear relationships, and causal effects) from large sets of data with

low information density to reveal relationships, dependencies and perform predictions of outcomes and behaviors.

Big data can also be defined as "Big data is a large volume unstructured data which cannot be handled by standard database management systems like DBMS, RDBMS or ORDBMS".

Big data can be described by the following characteristics:

Volume – The quantity of data that is generated is very important in this context. It is the size of the data which determines the value and potential of the data under consideration and whether it can actually be considered as Big Data or not.

The name 'Big Data' itself contains a term which is related to size and hence the characteristic.

Variety - The next aspect of Big Data is its variety. This means that the category to which Big Data belongs to is also a very essential fact that needs to be known by the data analysts. This helps the people, who are closely analyzing the data and are associated with it, to effectively use the data to their advantage and thus upholding the importance of the Big Data.

Velocity - The term 'velocity' in the context refers to the speed of generation of data or how fast the data is generated and processed to meet the demands and the challenges which lie ahead in the path of growth and development.

Variability - This is a factor which can be a problem for those who analyze the data. This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.

Veracity - The quality of the data being captured can vary greatly. Accuracy of analysis depends on the veracity of the source data.

Complexity - Data management can become a very complex process, especially when large volumes of data come from multiple sources. These data need to be linked, connected and correlated in order to be able to grasp the information that is supposed to be conveyed by these data. This situation, is therefore, termed as the 'complexity' of Big Data.

Big data analytics enables organizations to analyze a mix of structured, semistructured and unstructured data in search of valuable business information and insights.

1.4 IG DATA AND ITS MATTERS

It is difficult to recall a topic that received so much hype as broadly and as quickly as big data. While barely known a few years ago, big data is one of the most discussed topics in business today across industry sectors. This section has focus on what big data is, why it is important, and the benefits of analysing it.

1.5 DATA WE DISCUSS ABOUT

Organizations have a long tradition of capturing transactional data. Apart from that, organizations nowadays are capturing additional data from its operational environment at an increasingly fast speed. Some examples are listed here.

- Web data. Customer level web behavior data such as page views, searches, reading reviews, purchasing, can be captured. They can enhance performance in areas such as next best offer, churn modeling, customer segmentation and targeted advertisement.
- Text data (email, news, Face book feeds, documents, etc) is one of the biggest and most widely applicable types of big data. The focus is typically on extracting key facts from the text and then use the facts as inputs to other analytic process (for example, automatically classify insurance claims as fraudulent or not.)
- Time and location data. GPS and mobile phone as well as Wi-Fi connection makes time and location information a growing source of data. At an individual level, many organizations come to realize the power of knowing when their customers are at which location. Equally important is to look at time and location data at an aggregated level. As more individuals open up their time and location data more publicly, lots of interesting applications start to emerge. Time and location data is one of the most privacy-sensitive types of big data and should be treated with great caution.
- Smart grid and sensor data. Sensor data are collected nowadays from cars, oil pipes, windmill turbines, and they are collected in extremely high frequency. Sensor data provides powerful information on the performance of engines and machinery. It enables diagnosis of problems more easily and faster development of mitigation procedures.

1.6 DIFFERENCE OF BIG DATA AND TRADITIONAL DATA SOURCES

There are some important ways that big data is different from traditional data sources. In his book taming the big data tidal wave, the author Bill Franks suggested the following ways where big data can be seen as different from traditional data sources.

First, big data can be an entirely new source of data. For example, most of us have experience with online shopping. The transactions we execute are not fundamentally different transactions from what we would have done traditionally. An organization may capture web transactions, but they are really just more of the same transactions that have been captured for years (e.g. purchasing records). However, actually capturing browsing behaviour (how do you navigate on the site, for instance) as customers execute a transaction creates fundamentally new data.

Second, sometimes one can argue that the speed of data feed has increase to such an extent that it qualifies as a new data source. For example, your power meter has probably been read manually each month for years. Now we have a smart meter that automatically read it every

10 minutes. One are argue that it is the same data. It can also be argued that the frequency is so high now that it enables a very different, more in-depth level of analytics that such data is really a new data source.

Third, increasingly more semi-structured and unstructured data are coming in. Most traditional data sources are in the structured realm. Structure data are the ones like the receipts from your grocery store, the data on your salary slip, accounting information on the spreadsheet, and pretty much everything that can fit nicely in a relational database. Every piece of information included is known ahead of time, comes in a specified format and occurs in a specified order. This makes it easy to work with.

1.7 TENSOR-TRAIN DISCRIMINANT ANALYSIS

In this paper Seyyid Emre Sofuoglu et al,(2019) The rapid development of information technology is making it possible to collect massive amounts of multidimensional, multimodal data with high dimensionality in a diverse set of science and engineering disciplines. Although there has been a lot of recent work in the area of unsupervised tensor learning, extensions to supervised learning, feature extraction and classification are still limited. Moreover, most of the existing supervised tensor learning approaches are based on the Tucker model. However, this model has some limitations for large tensors including high memory and execution time costs. In this paper, we introduce a supervised learning approach for tensor classification based on the tensor-train model. In particular, we introduce two computationally efficient implementations of tensor-train discriminate analysis (TT-DA). The proposed approaches are evaluated on image classification tasks with respect to computation time, storage cost and classification accuracy.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

An outlier (also known as an anomaly) is a data point that is significantly different from the other data. Outlier detection aims to identify a small number of instances that deviate remarkably from the existing data. Detecting outliers in data is an important task, with many high-impact applications in areas such as health care. A number of anomaly detection approaches have been proposed. In fact, outlier detection plays an important role in identifying abnormal patterns and can be applied in different areas, including process control, environmental monitoring, traffic monitoring], and medical diagnosis.

There some vector-based methods for anomaly detection. MMSDE is proposed to identify the different health conditions of planetary gearbox proposed a fault diagnosis method based on AMMF and MHPE to detect the multiple gear fault types of planetary gearboxes.

Meanwhile, tensor representations can retain high-order correlations in the data. Therefore, it is necessary to retain the tensor data structure and design an outlier detection algorithm that can be applied to the original tensor data rather than vectorized data.

A new type of STM was also proposed, which was used for gait and action recognition. In recent years, unfolded matrices is used to construct nonlinear kernels. A tensor kernel is presented that retains tensor structures using a dual-tensorial mapping. It design a randomized support tensor machine, which is a novel structure-preserving kernel machine using randomized nonlinear features and a linear classifier to derive a highly scalable algorithm for tensorial anomaly detection.

A proposed support Tucker machines (STuMs), where weight parameters obtained using the Tucker factorization form a tensor. It presents a novel linear support higher-order tensor machine, which integrates the merits of a linear csupport vector machine and tensor rank-one decomposition.

It extends the concave-convex procedure-based transductive support vector machine to the tensor pattern and proposed a low-rank approximationbased transductive support tensor

2.2 PROPOSED SYSTEM

However, most of this work concentrates on learning tensor models based on CANDECOMP/PARAFAC (CP) factorization applied to explore tensor data. An important problem of tensor CP factorization is that the rank cannot be confirmed. If there are too many rank-one tensors, the included information may be noisy and redundant; if there are too few, the representation is incomplete. Hence, it is difficult for tensor CP decomposition to effectively use discriminative spatial information along each order. Another decomposition strategy is the Tucker decomposition, which is considered to be higher-order principal component analysis.

In the Tucker decomposition, each tensor is represented as the product of a core tensor and factor matrices along all orders. These matrixes can be thought of as the principal components, and the core tensor is the relationship among the principal components.

There are two advantages to Tucker decomposition. First, compared with the CP decomposition, which needs to evaluate the rank to approximate the initial tensor, we can obtain a more exact decomposition result for the tensor using Tucker decomposition. The other benefit is that we can reduce the dimensionality by adjusting the dimension of the core tensor.

To address the problems of big data outlier detection in IoT, we introduce Tucker decomposition into one-class support machines and propose the one-class support Tucker machine (OCSTuM).

IoT big data contains a large amount of useless information. These redundancies have a substantial impact on the performance of the algorithm. By extracting more information from the original IoT big data using minimal feature sets, we can save computing time and build a

better generalization for outlier detection in big data. The authors showed that feature subset selection affects the classification accuracy of machine learning algorithms.

In addition to feature selection, model parameters have a large impact on the performance of OCSTuM. Therefore, the appropriate model parameters can improve the detection accuracy of OCSTuM. Based on the above two considerations, we utilize a genetic algorithm (GA) to simultaneously select a feature subset of IoT sensor big

3. SYSTEM SPECIFICATION

3.1 SOFTWARE REQUIREMENTS:

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is useful in estimating cost, planning team activities and performing tasks throughout the development activity.

- Front End : PHP
- Back End : MYSQL
- Server : WAMP
- Operating System : Windows OS
- System type : :32-bit or 64-bit Operating System
- IDE : DREAMWEAVER
- DLL : Depends upon the title

3.2 FRONT END-PHP

Hypertext Preprocessor (or simply **PHP**) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994; the

PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications[9] and robotic drone control.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification.

As of August 2019, the majority of sites on the web using PHP are still on version 5.6 or older; versions prior to 7.1 are no longer officially supported by The PHP Development Team, but security support is provided for longer by third parties, such as Debian.

PHP development began in 1994 when Rasmus Lerdorf wrote several Common Gateway Interface (CGI) programs in C, which he used to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could be used to build simple, dynamic web applications. To accelerate bug reporting and improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" on the Usenet discussion group comp.infosystems.www.authoring.cgi on June 8, 1995. This release already had the basic functionality that PHP has today. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl, but was simpler, more limited and less consistent.

Early PHP was not intended to be a new programming language, and grew organically, with Lerdorf noting in retrospect: "I don't know how to stop it, there was never any intent to write a programming language [...] I have absolutely no idea how to write programming language, I just kept adding the next logical step on the way." A development team began to form and, after months of work and beta testing, officially released PHP/FI 2 in November 1997.

PHP 5 parser in 1997 and formed the base of PHP 3, changing the language's name to the recursive acronym PHP: Hypertext Preprocessor. Afterwards,

Many high-profile open-source projects ceased to support PHP 4 in new code as of February 5, 2008, because of the GoPHP5 initiative, provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5.

Over time, PHP interpreters became available on most existing 32bit and 64-bit operating systems, either by building them from the PHP source code, or by using pre-built binaries. For PHP versions 5.3 and 5.4, the only available Microsoft Windows binary distributions were 32-bit IA32 builds, requiring Windows 32-bit compatibility mode while

using Internet Information Services (IIS) on a 64-bit Windows platform. PHP version 5.5 made the 64-bit x86-64 builds available for Microsoft Windows.

Official security support for PHP 5.6 ended on 31 December 2018, but Debian 8.0 Jessie will extend support until June 2020.

3.3 PHP 6 and Unicode

PHP received mixed reviews due to lacking native Unicode support at the core language level. In 2005, a project headed by Andrei Zmievski was initiated to bring native Unicode support throughout PHP, by embedding the International Components for Unicode (ICU) library, and representing text strings as UTF-16 internally. Since this would cause major changes both to the internals of the language and to user code, it was planned to release this as version 6.0 of the language, along with other major features then in development.

However, a shortage of developers who understood the necessary changes, and performance problems arising from conversion to and from UTF16, which is rarely used in a web context, led to delays in the project. As a result, a PHP 5.3 release was created in 2009, with many non-Unicode features backported from PHP 6, notably namespaces. In March 2010, the project in its current form was officially abandoned, and a PHP 5.4 release was prepared containing most remaining non-Unicode features from PHP 6, such as traits and closure rebinding. Initial hopes were that a new plan would be formed for Unicode integration, but as of 2014 none had been adopted.

3.4 PHP 7

During 2014 and 2015, a new major PHP version was developed, which was numbered PHP 7. The numbering of this version involved some debate. While the PHP 6 Unicode experiment had never been released, several articles and book titles referenced the PHP 6 name, which might have caused confusion if a new release were to reuse the name. After a vote, the name PHP 7 was chosen.

The foundation of PHP 7 is a PHP branch that was originally dubbed PHP next generation (phpng). It was authored by Dmitry Stogov, Xinchen Hui and Nikita Popov,[49] and aimed to optimize PHP performance by refactoring the Zend Engine while retaining near-complete language compatibility. As of 14 July 2014, WordPress-based benchmarks, which served as the main benchmark suite for the phpng project, showed an almost 100% increase in performance. Changes from phpng are also expected to make it easier to improve performance in the future, as more compact data structures and other changes are seen as better suited for a successful migration to a just-in-time (JIT) compiler.

Because of the significant changes, the reworked Zend Engine is called Zend Engine 3, succeeding Zend Engine 2 used in PHP 5.

4. BACK END SOFTWARE – MYSQL

4.1 MYSQL INTRODUCTION

The MySQL® database has become the world's most popular open source database because of its consistent fast performance, high reliability and ease of use. It's used on every continent -- Yes, even Antarctica! -- by individual Web developers as well as many of the world's largest and fastest-growing organizations to save time and money powering their high-volume Web sites, business-critical systems and packaged software -- including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, We Tube, and Zappos.com.

Not only is MySQL the world's most popular open source database, it's also become the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) MySQL runs on more than 20 platforms including Linux, Windows, Mac OS, Solaris, HP-UX, IBM AIX, giving we the kind of flexibility that puts we in control.

Whether we're new to database technology or an experienced developer or DBA, MySQL offers a comprehensive range of certified software, support, training and consulting to make we successful.

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP based web application or as a standalone database server.

Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are however limits to how far performance can scale on a single server, so on larger scales, multi-server MySQL deployments are required to provide improved performance and reliability.

A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations.[18] The

master server synchronizes continually with its slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime.

Further improvements in performance can be achieved by caching the results from database queries in memory using memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

4.2 HTML

HTML remains for Hyper Text Markup Language. It is a basic content designing dialect used to make hypertext records. It is a stage free dialect not at all like most other programming dialect. HTML is impartial and can be utilized on numerous stage or desktop. It is this component of HTML that makes it mainstream as standard on the WWW. This adaptable dialect permits the making of hypertext connections, otherwise called hyperlinks. These hyperlinks can be utilized to unite reports on diverse machine, on the same system or on an alternate system, or can even indicate purpose of content in the same record. HTML is utilized for making archives where the accentuation is on the presence of the record. It is likewise utilized for DTP.

The records made utilizing HTML can have content with diverse sizes, weights and hues. It can also contain graphics to make the document more effective. Hyper Text Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.[1] Web browsers can read HTML files and render them into visible or audible web pages.

HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language. in the form of HTML elements consisting of tags enclosed in angle brackets (like). Browsers do not display the HTML <html> tags and scripts, but use them to interpret the content of the page.

HTML can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and lawet of text and other mate

5. METHODOLOGY

5.1 PROBLEM DEFINITION

The medical reports and the records are stored in to the database server which is stored with quiet insecure manner. The details of the patients are managed in a cloud server and have a chance of data breach. So the administrator of the records is in urge to maintain the

security of the data which causes a serious issue in the medical environment. So the major problem is to maintain the patient records in a cloud are not up to the level.

5.2 IMPLEMENTATION

System implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successfully system and in giving confidence on the new system for the user that it will work efficiently and effectively.

- Cloud Framework
- Data Encryption
- Cloud Data Storage Using Stegnography
- Key Generation
- Track Intruder
- Abnormal Alert

5.3 MODULE DESCRIPTION

5.3.1 CLOUD FRAMEWORK

The Cloud Computing Framework (CC) is a promising paradigm for on-demand access to a shared set of resources such as networks, servers, storage, and services. Cloud Computing allows us to access those shared resources in an efficient and convenient manner without the need to maintain hardware resources. Providing efficient scheduling schemes of virtual machines can significantly improve the energy efficiency of Cloud data centers when dealing with resource intensive applications

5.3.2 DATA ENCRYPTION

The medical reports stored in the database are in an encrypted format which ensures the data security and the key has been provided to the physician and as well as the patient side. Due to encrypted data format it is quiet difficult to access

5.3.3 CLOUD DATA STORAGE USING STEGNOGRAPHY

To increase the security level of the data that has been stored in the cloud Stegnography technique has been used which merges the patients' medical reports into an image and hide for security issues, which prevents the data from third party access.

5.3.4 KEY GENERATION

Once the data has been uploaded by the doctor security keys will be generated to the doctor and as well as to the patients. The physician can access the respective patients' reports with the public key that has been generated randomly from the cloud server which provides the privacy for every individual patients involved in the system.

5.3.5 ABNORMAL ALERT

Incase of occurrence of abnormal condition in the patients' health conditions which was monitored by the sensors will be automatically updated to the cloud server and the respective physician and as well the guardian of the concerned patient will receive the status intimation for further processing which enhances the efficiency of the system in a better way.

6. RESULTS AND DISCUSSION

In our experiments, we evaluate the results with accuracy, recall, FP and precision, which are typical performance metrics to evaluate models used in machine learning or deep learning. In the following, TP refers to “true positive”, TN refers to “true negative”, FP refers to “false positive” and FN refers to “false negative”. Accuracy (ACC) indicates the proportion of all requests that are correctly detected over all the data as follows:

ACCURACY

ACC=$TP+TNTP+FP+TN+FN$/100

TRUE PRECISION RATE

Recall (TPR) is the ratio of real attacks that are detected as anomalous over all attacks as follows:

TPR=$TPTP+FN$

FALSE PRECISION RATE

The FPR rate is the ratio of normal requests that are detected as attacks over all normal requests as follows

FPR=$FPFP+TN$

PRECISION

Precision is defined as the ratio of anomalous predictions that are correct as follows:

Precision=$TPTP+FP$

Particularly, we define the rate of real normal requests that are detected as normal over all normal requests (DRN) as follows:

DRN=$TNFP+TN$

In healthcare scenarios, where different types of data have to been managed and cross-related. Some models and techniques for health data visualization have been presented in literature. However, they do not satisfy the visualization needs of physicians and medical personnel. Here we present a new graphical tool for the visualization of health data that can be easily used detect anomaly detection in data stored. The tool is very user friendly, and allows physician to maintain the privacy of the data.

The Advanced Encryption Standard (AES), also known by its original name Rijndael is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.AES is a subset of the Rijndael cipher developed by two Belgian cryptographers, Vincent Rijmen and Joan Daemen, who submitted a proposal to NIST during the AES selection process. Rijndael is a family of ciphers with different key and block sizes.

As the preprocessing measures are much lower when compared with the other algorithms the implementation of CNN in this procedure analysis the entire data that has been involved in the system to gather the complete reference. As CNN considers the entire sequence in a layer by layer format every analysis report can be acquired with better accuracy i.e.: the moisture level in the soil fertility has been predicted with CNN in an accurate manner, as it considers each and every layer. As we use CNN, the entire collected datasets are analyzed in an efficient manner and each and every mentioned errors and predicted values are deeply gathered to attain better comparison and security results and values. With the help of those compared values accuracy can be obtained to its core.

Table.5.1 Accuracy Level Prediction Proposed Method

Monitoring & Privacy	0.9516	**T P**	115	95.16%
	0.9358	**TN**	6	93.58%
Anomaly Detection & Revoking	0.9583	**FP**	38	95.83%
	0.9576	**F N**	73	95.86%
Accuracy				95.08%

Fig.5.5 CNN and AES Prediction Levels

7.CONCLUSION

In this article, we introduced the OCSTuM and GA-OCSTuM methods for detecting outliers in IoT sensor big data. More specifically, in contrast to methods using the CP decomposition, which needs to evaluate the rank to approximate the initial tensor, we can obtain a more exact decomposition for the tensor using the Tucker decomposition. The other benefit is sthat we can reduce the dimensionality by adjusting the dimension of the core tensor. Therefore, the Tucker factorization is used to compress the attributes of each sample in large-scale data. OCSTuM was proposed to detect outliers in the sensor big data. OCSTuM optimization uses time-consuming iterative techniques that are the main cause the inefficiency of OCSTuM to solve solution. Hence, GA-OCSTuM was proposed for simultaneous feature selection and parameter optimization for anomaly detection IoT sensor big data. The proposed method can effectively exploit the principal components of sensor big data to improve the detection performance. The experimental results showed that, compared with vector-based anomaly detection methods, the detection accuracy of the proposed algorithm are better.

8. FUTURE ENHANCEMENT

In future works, for each measurement, we plan to implement a non-linear regression algorithm able to quantify the anomaly detection with high precision. Furthermore, the development of a Machine Learning techniques would help us to create a model capable of predicting the relative measurement values in order to send alarms to the concerned data center owner when risky conditions occur i.e.: the intruder or the user from the other sources who tries to access the data without proper authentication.

9. REFERENCES

[1] Dagstuhl perspectives workshop: Tensor computing for internet of things.

[Online]. Available: www.dagstuhl.de/en/program/calendar/semhp/?semnr=16152 (last accessed: May 21, 2016).

[2] Statista. Internet of things (iot): number of connected devices worldwide from 2012 to 2020 (in billions). [Online]. Available:

http://www.statista.com/statistics/471264/iot-number-of-connected-dev icesworldwide/

[3] Deng X, Jiang P, Peng X, et al. Support high-order tensor data description for outlier detection in high-dimensional big sensor data [J]. Future Generation Computer Systems, 2018, 81: 177-187.

[4] D. Hawkins. Identification of outlier, Chapman and Hall, 1980.

[5] Kriegel H P, Zimek A. Angle-based outlier detection in high-dimensional data[C] //Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008: 444-452.

[6] Angiulli F, Basta S, Pizzuti C. Distance-based detection and prediction of outliers. IEEE transactions on knowledge and data engineering, 2006, 18(2):

145-160.

[7] Jiang Y, Zeng C, Xu J, et al. Real time contextual collective anomaly detection over multiple data streams. Proceedings of the ODD, 2014: 23-30.

[8] Mori J, Yu J. Quality relevant nonlinear batch process performance monitoring using a kernel based multiway non-Gaussian latent subspace projection approach. Journal of Process Control, 2014, 24(1): 57-71.

[9] Engle M A, Gallo M, Schroeder K T, et al. Three-way compositional analysis of water quality monitoring data. Environmental and Ecological Statistics, 2014, 21(3): 565-581.

[10] Fanaee-T H, Gama J. Event detection from traffic tensors: A hybrid model.

Neurocomputing, 2016, 203: 22-33.