IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ADVANCED CLOUD SECURITY: EMPOWERING DATA PROTECTION THROUGH DENSE BELIEF NETWORKS AND RK-AES ALGORITHM

Jamal Kutty. K [1], Mohamed Ashraf. K [2]

[1]Lecturer,[2] Lecturer

[1]Department of Electronics Engineering, AKNM Government Polytechnic College
Tirurangadi, Malappuram, Kerala, India

[2]Department of Electronics Engineering, IPT And Government Polytechnic College
Shoranur, Palakkad, Kerala, India

***Abstract**:* As the different techniques such as Artificial Intelligence, Internet of Things, Blockchain revolutionized among today's generation, there is also an equally revolutionized technology which is Cloud Computing that is been used by millions of users for saving data, transferring data, and many more. As everything is getting advancing, there is also an important factor that every user's or say customers that are concerned with i.e., security. In data centers of massively scalable, the cloud services and data are residing and from everywhere it can be accessed. The increase of cloud customers has sadly been accompanied by an increase in malicious activity in the cloud. Almost every day more and greater vulnerabilities are discovered and published in a new security advisory. For different purposes Cloud surfing is done by millions of users; so that, they want highly persistent and safe services. So, in this paper we propose an effective enhanced security-based cloud computing using Cryptography and also Deep Learning (DL) approach was, Dense Belief Network is used as classifier layer for classifying various intrude data and then finally passed to Advance Encryption Standard before it is being launched to the cloud or ensuring data confidentiality and security. We also evaluated our proposed model (AES-RK-DBN) concerning other models such as LSTM, DenseNet, Resnet, VGG, CNN underperformance measures such as Accuracy, Specificity, Detection Rate, Sensitivity, and mostly Security in which our model gives higher satisfaction in terms of accuracy and security.

Index Terms - Cloud Computing, Deep Learning, Deep Belief Network, Encryption, Internet of Things, RK-AES Algorithm

I. INTRODUCTION

Recently the cloud computing, cluster computing, and grid computing are the computing paradigms on which computing to services transformation are delivered and customaries such as typical utilities (electricity, water, and fuel) depend on, IT (information technology) services can attract and transform by the Cloud computing as a utility. Operation costs, as well as the capital, outlays, are reduced by this innovative idea. In the IT sector, the fastest developing field is cloud computing because of this potential capacity. By the use of the service provider's hardware and software facility over the internet, application as service delivery is provided by the Cloud computing in which it is defined, which can be either known as Platform as a Service (PAS), Software as a Service (SaaS) or Infrastructure as a service (IaaS) [1]. The cloud is formed by the software and hardware part which is normally known as a public cloud whereas per the use manner in pay the services are offered, under the utility computing it comes. In which full access to the business/organization to which the facility is availed is referred to as the private cloud while only a limited access to the general public. The facility of utility computing and SaaS's together is the Cloud computing, where either the people can be providers or users of the facility of former aforesaid hence excluded the data centers (medium and small). Rather than targeting the individual computers for a vast group, the computing world transformation is towards the development of software as services [2].

For availing the services, the service providers charge the customers were over a network the services of cloud computing are accessed which offers business applications capacities. All the IT functionalities are delivered by the technology of Cloud computing and the computing's upfront costs are reduced dramatically to the companies which may give the cutting-edge services [3]. Around the globe for cloud computing the data centers have been launched especially by the providers like Microsoft, International Business Management (IBM), Salesforce, Google, and Amazon as a part of reliability, Total Quality Management (TQM), and redundancy [2]. The Advanced Research Project Agency Network (ARPANET) implementation was the vision of the 21st century, which is considered as utility computing’s major milestone towards the aim of success, which became popular as internet and World Wide Web (WWW) [4]. To the user requirements with real-time response, the convergence of business agility as well as IT efficiency is combined in Cloud computing [3]. To cloud computing in counterpart, grid computing and cluster computing are widely explored other computing paradigms. From the grid's perspective of electrical power with inspirations sharing resources between resources of geographically distributed is enabled in Grid computing. As computing resources of a single integrated group of computers work in the networks of interconnected and parallel which is involved in Cluster computing.

Inside cloud computing for offering security assurance, formed a non-profit organization known as Cloud Security Alliance, the use of best practices to promote. In the cloud the companies and individuals more and more information is stored, issues are commencing to develop about simply how protected an environment it is. During cloud engineering, the cloud service providers (CSP) face challenges, security issues, and requirements which are discussed in this paper, and also several options to mitigate them. The market can thrive and evolve because some form of standardization is needed for it (e.g. Open Virtualization Format (OVF), Information Technology Infrastructure Library -ITIL). No matter which dealer offers cloud services, with each other to communicate and interoperate, clouds should be allowed by the Standards [3]. In virtual machines to be run, for the distribution and packaging of software, extensible, efficient, portable, secure, open, platform-independent and as vendor OVF standard is highly recommended by it. A very difficult task that makes effective data utilization is before outsourcing to encrypt the sensitive data for data privacy protection. We have to think about some of the addressed issues and their remedy in the cloud for retaining security. No matter which seller offers cloud services, with every other to communicate and interoperate the clouds should be allowed by some agreements and standards. We can keep our information safely because the technique of encryption is introduced.

Concern among big cooperate companies about handling their operations through another firm and bankruptcy of cloud providers especially in a shrinking economy. Followed by reliability and performance among the IT executives, security is also a serious concern [16]. The cloud services acceptability is reduced because of the Lack of standards mainly International Organizations for Standards are still missing. Throughout the European Union (EU) the client's interest is safeguarded by implementing standards and being checked, where the most common example is the EuroCloud launch. The structure of cloud computing is demonstrated in fig 1.

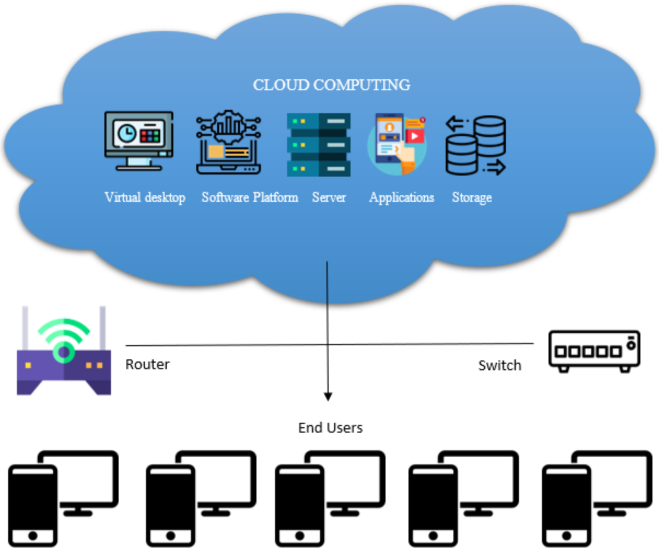


Figure. 1: basic architecture of cloud computing model

Organization of paper: As we went through an introductory part in Section 1, the remaining sections are as follows: Section 2 depicts related works that is been so far done by researchers, section 3 brings methodologies of the proposed model AES-RK-DBN, Section 4 gives the simulation results of the proposed model and finally the paper concludes with section 5. Reference is given in the end part**.**

II. **Related Works**

With the advancement in cloud computing, the training dataset, [42] of the deep learning model, privacy-preserving model [7] and sharing file's flexible access control [55], security robustness [54], authentication of protocols [53], [41], data mining [30], cloud storage [38]–[40], [37] and cloud architecture [36] are which the security confronting problems in the cloud, considering these there has been numerous existing works. In the cloud, the schemes of privacy-preserving cryptographic's existing works are based totally on secure multi-party computations. (a) training without the help of the cloud server and (b) training on the cloud server are the two different domains where the cryptographic algorithms are based on.

For example, based on back-propagation known as BPNN in which the algorithm of privacy-preserving of a two-party distributed is introduced by Chen et al. [43]. While between two parties over the arbitrarily portioned data, a similar BPNN is introduced by Bansal et al. [44]. Computational operations of secure two-party are supported by them using the ElGamal scheme. Without the leakage of sensitive information about any data providers, the training of neural networks is enabled by these algorithms. To preserve intermediate results and data privacy, the scheme of fully homomorphic encryption is applied by them. The data vertically partitioned are conducted in the algorithms they proposed, which means every data provider has a feature vector subset. Although, for multiple parties, there is a lack of solutions to conduct deep learning of collaborative datasets arbitrarily partitioned with each. By the use of cloud computing resources, to this problem, a practical solution is suggested by Yuan et al. [45]. To the overhead of expensive communication, the multi-party scenario to which the algorithms of application of [43]– [45] may lead.

CryptoNet is introduced by Bos et al. [46] which is a model of novel privacy-preserving, to make sure that the datasets stay private which allows the data of homomorphically encrypted to be outsourced by the data provider. A model of feed forwarding pretrained is the CryptoNet, by data providers to the cloud, the outsourced encrypted data on which it is applied. CryptoNet is a pre-trained feedback model used by data providers on encrypted data abstracted to the cloud. It uses the completely homomorphic encryption technique [47] in order to assess a deep neural network of two coevolutionary layers and two fully connected layers. The ongoing effort to protect data privacy resulted to the suggestion from the CryptoNet

[10] that the deep neural network can receive these services on completely homomorphic encrypted data without exposure of sensitive data to either the supplier of the cloud service or other data providers. Based on the secret-sharing model of Shamir with the encryption of leveled homomorphic, the convolutional neural network's combination is CryptoDL.

By the use of the method of stochastic gradient descent also known as SecureML for the training of privacy-preserving neural networks efficient and new protocols are also introduced by Li et al. [18]. Among the servers of two non-colluding, the private data of the providers are distributed on which in the model of two-server this algorithm falls. By the use of the computations of secured two-party, on the collaborative data, performs the training of models of different deep neural networks. On the shared decimal numbers, secure arithmetic computations are also supported by the SecureML. Table 1 shows the comparative review of the existing models.

Table 1: the comparative review of the existing models.

Authors	Methods	Accuracy (%)
Chen et al. [43]	BPNN	82
Bansal et al. [44]	BPNN	88.25
Bos et al. [46]	CryptoNet	92
Li et al. [18]	SecureML	92.25

III. METHODOLOGY

In academia and industry, an excellent deal of interest is created by cloud computing. In cloud computing one of the most famous problems is the security of data. Hence, in this article, we proposed a suitable method for data storage and intrusion detection. Security of text documents in the cloud environment is mainly focused on in this article. In the suggested part, carried out three major operations, are Encrypting data storage, Data preprocessing, and Intrusion Detection. By the use of a proposed algorithm, the process of Encryption data storage and intrusion detection's overall structure is illustrated. RK-AES algorithm is used for filtering purposes. Figure 2 shows the block diagram of AES-RK-DBN.

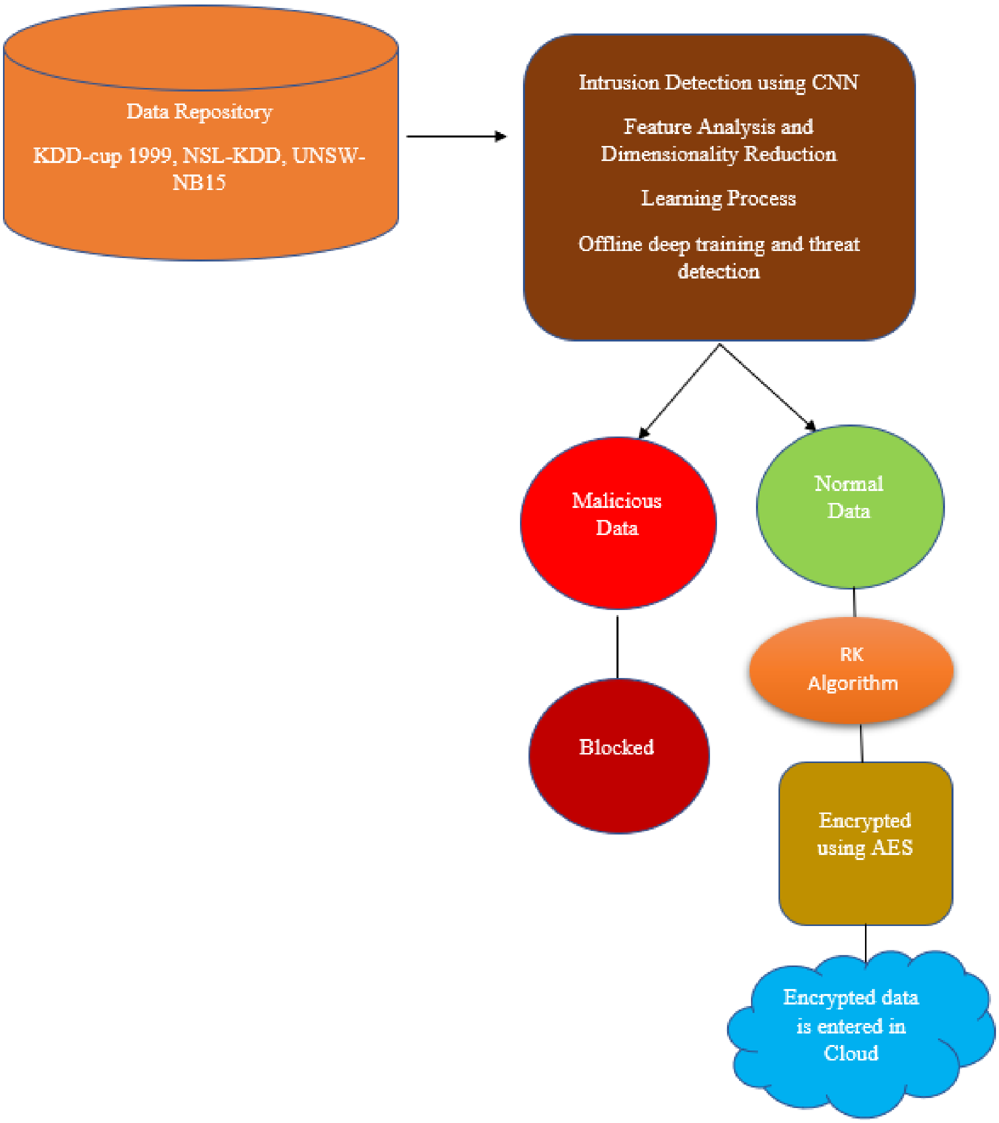

Data Repository
KDD-cup 1999, NSL-KDD, UNSW-NB15
Intrusion Detection using CNN
Feature Analysis and Dimensionality Reduction
Learning Process
Offline deep training and threat detection
Malicious Data
Normal Data
RK Algorithm
Blocked
Encrypted using AES
Encrypted data is entered in Cloud

Figure. 2: block diagram of AES-RK-DBN

3.1 Dataset Collection

The Datasets used are

1) KDDcup 1999 Dataset: For the network model of intrusion detection as a benchmark the KDDcup 1999 dataset [16] is broadly utilized. In the dataset 41 features are contained in every record and either it is labeled as an attack of a specific type or normal. The attacks of 22 types are contained in the training dataset, while additionally 17 types are contained in the testing dataset.

2) NSL-KDD Dataset: In both the testing and training dataset of the KDDCup 1999 dataset a big amount of redundant records is contained to resolve this inherent problem in [17] the NSL-KDD Dataset was introduced. 41 features are there in every traffic sample. In the dataset, Probe, U2R, R2L, and DoS attacks are the four categories in which attacks are classified. 24 types of attack are included in the training dataset, while 38 types of attack are contained a testing dataset.

3) UNSW-NB15 Dataset: Namely Worms, Shellcode, Reconnaissance, Generic, Exploits, DoS, Backdoors, Analysis, and Fuzzers are the 9 families of attacks this dataset has. To generate features of a total of 49, developed 12 algorithms and utilized the tools of network monitoring like Bro-IDS and Argus. From the attacks of different types, 82,331 records are in the testing dataset and 175,340 records are in the training dataset.

3.2Deep Learning Model

Explained, in the cloud system, how the cyberattacks are detected by the learning model and for intrusion detection the deep learning model is introduced by us in this section. Figure 3 shows the architecture of the Deep Learning model.

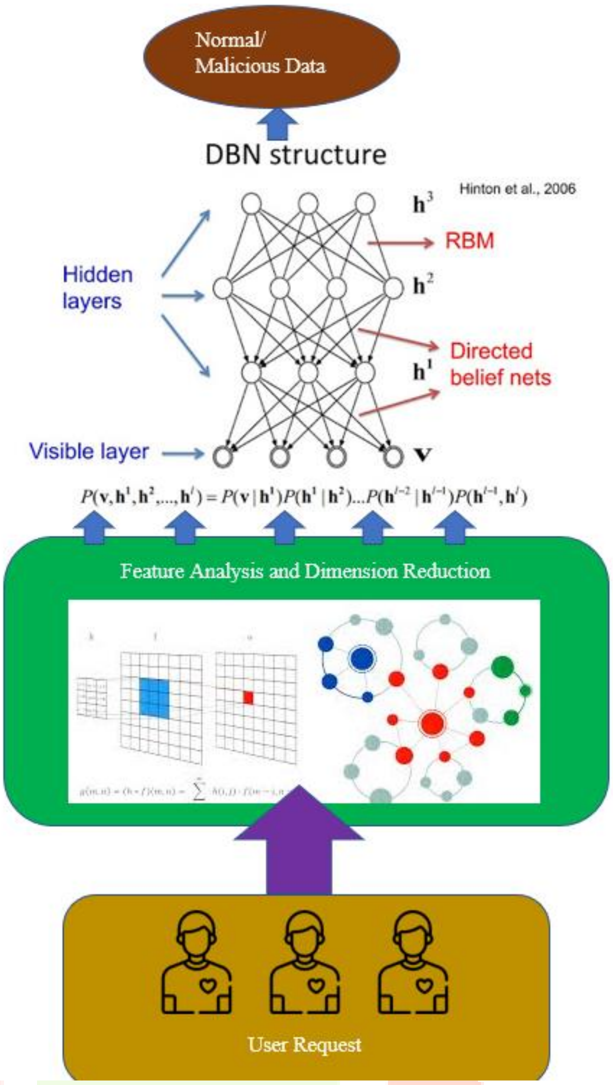


Figure.3: deep learning model

The phases of the Deep Learning Model are given below:

3.2.1Dimension Reduction and Feature Analysis

Features Analysis: In the model of deep learning first step is the feature analysis. Extract the features and analyze the features are the purpose of this step. Malicious packets of different kinds may additionally have special features, from the regular ones which are different, Whether or not a packet is malicious can be determined by fetching and examining the packet's abnormal attributes. e.g. to discover DoS attacks [13] important features are IP packet entropy, source bytes, and percentage of packets with errors.
Dimension Reduction: With various features, many attributes are contained in data packets. E.g. 41 features are contained in every record in the NSL-KDD dataset [17] and KDDcup 1999 dataset [16]. Although, for intrusion detection, all 41 features are not beneficial. Some of the features are redundant and irrelevant, resulting in a performance degrading and process of long detection. Hence, the dataset's most necessary information is maintained by deciding features, which is vital to reducing the complexity of computation and for the increased learning process accuracy.
Principal Component Analysis (PCA) is an efficient method, to emphasize variation in machine learning which is utilized and, in a dataset, strong patterns are determined. Minimize the dataset's dimensionality is the core concept of PCA of which containing interrelated variables of a big amount, while in the dataset [14], the presented variation is preserved as much as possible. So that, for considered datasets to decrease dimensions, the PCA is adopted in this article.
Mathematically, to an r-dimensional space, from an n-dimensional space, a dataset is mapped by the PCA, where $r \leq n$, the projection's residual sum of squares (RSS) to reduce. The projected dataset's [14] covariance matrix's maximization is equal to this. Two important properties were the new domain dataset has i.e., according to their information's importance ordered the dimensions and anymore, no correlation, the various dimensions of the data have. With n various variables and m observations, we can define X as an $(m \times n)$ matrix. Then, the C which is a covariance matrix is given as:

$$\mathrm{C} = \frac{1}{n-1}\mathrm{X}^{\top}\mathrm{X} \tag{1}$$

C can be diagonalized as follows [15], Since C is a symmetric matrix:

$$C = VLV^{\top} \tag{2}$$

Where eigenvectors matrix is V, and L = $diag(\lambda_1 \ldots \lambda_p)$

In decreasing order, it is an eigenvalues diagonal matrix. To perform PCA, if the singular value decomposition (SVD) is used. The decomposition as follows [15] is obtained:

$$X = U\Sigma V^{\top} \tag{3}$$

Where orthonormal matrices are V and U, meaning that $U^T U = UU^T = I$ and $V^T V = VV^T = I$ and $\Sigma = diag(s_1 \ldots s_n)$ is a singular value s_i. diagonal matrix. Then the following results are derived:

$$C = \frac{1}{n-1}X^{\top}X = \frac{1}{n-1}(V\Sigma U^{\top})(U\Sigma V^{\top}) \tag{4}$$

$$= \frac{1}{n-1}V\Sigma^2 V^{\top}$$

has principal directions that singular vector V is (4) implies and to the eigenvalue λ_i, singular value s_i is related, to covariance matrix C via $\lambda_1 = {s_i}^2/(n - 1)$. Thus, the principal components (PCs) can define as follows [15]:

$$P = XV = U\Sigma V^T V = U\Sigma \tag{5}$$

where the PCs are the matrix P's columns and the matrix V is known as loading matrix, for each PC the variable's coefficients linear combination is contained. While preserving the important dimensions of the data, to r-dimensional, from n-dimensional that the dataset needs to project. In different words, the smallest valuer has to find such that the following circumstance holds:

$$\frac{\sum_{i=1}^{\top}\lambda_i}{\sum_{j=1}^{\top}\lambda_j} \geq \alpha \tag{6}$$

To r-dimensional after reducing the input data dimension, needs to be reserve the α which is the percentage of information. We can have a look at that the PCs are chosen by PCA i.e., important features, the variance α is maximized by that.

3.2.2. Learning Process

As demonstrated in Fig. 2 some hidden, input, and output layers are the three layers included in the process of learning. In the input layer, input data will be utilized for the refined features. Whether or not the packet is malicious or normal is determined after the learning process. As shown in Figure. 2, SoftMax regression steps, pre-learning, and deep learning are the three main steps involved in the learning process.

a. Pre-learning Process: To transform actual values Gaussian Binary Restricted Boltzmann Machine (GRBM) is used in this step, i.e., input layer's input data, into binary codes, and then in the hidden layers which will be utilized. j hidden units and i visible units were GRBM has. In advance, it is pre-defined the number of hidden units and after reducing the dimension, as the number of features, defines the number of visible units (i.e., the no. of neurons). The GRBM's energy function is described as:

$$E(v,h) = \sum_{i=1}^{I}\frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{i=1}^{I}\sum_{i=1}^{J} w_{ij}h_j\frac{v_i}{\sigma_i} - \sum_{i=1}^{J} b_j h_j$$

(7)

Where hidden vector is hand visible vector is v. To visible and hidden units, a_i and $b_j$are biases respectively.

Between the hidden and visible units, the connecting weight is the w_{ij}, and with Gaussian visible unit v_i, associated is the standard deviation σ_i. Through the energy function, to each possible pair of a hidden and a visible vector, a probability is assigned by the network. Defused the probability as follows:

$$p(v,h) = \frac{e^{-E(v,h)}}{\Sigma_{v,h}e^{-E(v,h)}} \tag{8}$$

We can derive, from (8), to a visible vector v, the network assigned a probability that is as follows:

$$p(v) = \frac{\Sigma_h e^{-E(v,h)}}{\Sigma_{v,h}e^{-E(v,h)}}$$

(9)

In the training data's log probability for performing the stochastic steepest descent the learning update rule can derive from the probability p(v) as below:

$$\frac{\partial log\, p(v)}{\partial w_{Ny}} = \langle\frac{1}{\sigma_i}v_i h_j\rangle_{data} - \langle\frac{1}{\sigma_i}v_i h_j\rangle_{model}$$

(10)

$$\Delta\omega_{i,j} = c\left(\langle\frac{1}{\sigma_i}v_i h_j\rangle_{data} - \langle\frac{1}{\sigma_i}v_i h_j\rangle_{model}\right)$$

Where the learning rate is the c and by the subscript a distribution specified under which the expectation is denoted using a (.) that follows [22]. Getting a $\langle v_i h_j\rangle_{model}$ the unbiased sample is difficult because in a GRBM, between the visible units and the hidden units there is no connection. Hence, to tackle this problem can apply the sampling strategies. Normally, at the visible units any random state we can start, and alternately Gibbs sampling can perform. Using equation (11), parallelly updating all the hidden units is involved in the alternating Gibbs sampling of every iteration followed by equation (12), parallelly updating all visible units.

$$p\left(h_j = 1 \mid \mathrm{v}\right) = \mathrm{sig}\,m\left(b_j + \sum_i w_{ij}\frac{v_i}{\sigma_i}\right)$$

(11)

$$p(v_i \mid \mathrm{h}) = \mathcal{N}\left(v_i \mid a_i + \sum_j h_j w_{ij}, \sigma_i^2\right) \qquad (12)$$

Where the sigmoid function is the $sigm(x) = 1/(1 + exp(-x))$ and a Gaussian probability dainty function is denoted by $N(-|\mu, \sigma_i^2)$ with standard deviation σ and mean μ.

a. ***Deep Leaning Step***: To adjust the neural network weights, in sequence a series of learning processes is performed which is included in this step. Between the two successive layers, via Restricted Boltzmann Machine (RBM) in the hidden layers pertain every learning process, particularly a Markov random field type is an RBM. It has an architecture of two-layer in which the hidden binary stochastic units $h \in \{0,1\}^F$, the visible binary stochastic units $v \in \{0,1\}^D$ are connected. Here, the numbers of hidden and visible units are F and D respectively. Then, by [22] can calculate the energy of state {v, h}:

$${}_{ij}v_i h_j - \sum_{i=}^{D} a_i v_i - \sum_{j=1}^{F} b_j h_j \qquad (13)$$

Similarly, where parametersw_{ij}, a_i and b_j are defined as in (7).

One is the single variable's conditional probability (e.g., $p\left(h_j = 1 \mid v\right)$) with the sigmoid activation function as a neuron's firing rate it can be interpreted as follows [22]:

$$\left(\sum_{i=1}^{D} w_{ij}v_i + b_j\right)$$

(14)

$$= \mathrm{sigm}\left(\sum_{i=1}^{F} w_{ij}h_j + a_i\right)$$

(15)

For the weights of the RBM the learning update rule can be derived similar to the pre-learning step, as follows:

$$h_j\rangle_{data} - \langle v_i h_j\rangle_{model}\Big)$$

(16)

where c as the learning rate

b. Softmax Regression Step: For the packet classification, as the softmax regression's input (at the output layer), the last hidden layer's output will be used i.e., x. Into the classes of M= (K + 1) classifies the packet. Attacks of all types are denoted by K. Mathematically, class i is an output prediction of Y's probability. is decided by.

$$\mathrm{ftmax}_i(\mathrm{Wx} + \mathrm{b}) = \frac{e^{W,\mathrm{x}+h_2}}{\sum_j e^{W,\mathrm{x}+b_j}} \qquad (17)$$

Where a bias vector is b and between the output layer and the last hidden layer W is a weight matrix. Then, the class is the model's prediction y_{pd} whose probability is maximal, specifically:

$$y_{pd} = arg\ argmax_i[p(Y = i|x, W, b)], \forall i \in \{1,2, \dots, M\} \qquad (18)$$

3.2.2 Online Threat Detection and Offline Deep Training

Phases of fine-tuning and pre-training are the two phases contained in deep training.

1) Pre-training: For training, from the Internet, only the unlabeled data which is easy and cheap to collect is required in this phase. By the use of a group of simple sub-models, to study a complicated model an effective way is introduced by the authors in [18] which are sequentially learned. To have various data representations in the sequence each sub-model is allowed by the learning algorithm of greedy layer-wise. To generate output vectors, on its input vectors a non-linear transformation is performed by the sub-model. In the sequence as the next sub-model's input that will be utilized. For every layer [18], [19], [20] as the building blocks with RBMs, can be applied for each layer's principle of unsupervised training in greedy

layer-wise. With the gradient [21] approximation by the use of CD using Gibbs sampling execute our training process.
2) Fine-tuning: For fine-tuning a set of labeled data available is used. At a time for one-layer sensible set of weights, we have after the phase of pre-training. So that for better discrimination the model can be fine-tuned by the use of bottom-up back-propagation.

With the trained weights a model of deep learning will acquire after the completion of offline deep training. In an online fashion to detect the malicious packets, the module of attack detection will be implemented in this model of learning.

3.3 RK-AES Model

Once the Deep Learning model is classified data into malicious or non-malicious, the malicious data is blocked. The non-malicious data is passed to RK-AES Model for encryption.

3.3.1 AES Algorithm

In 2001 the National Institute of Standards and Technology (NIST) published the Advanced Encryption Standard (AES) [26]. For the both decryption and encryption process, a single key is utilized by the AES, which is a symmetric block cipher. 128 bits sequences were contained in every output and input of the AES algorithm. 128, 256, or 192 bits were contained in the key utilized by this algorithm. On which in 8-bit bytes the AES operates. By the use of the following polynomial representation, as an infinite field element, interpreted these bytes:

$$b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots + b_1x + b_0 = \sum_{i=0}^{n-1} b_1x^i \qquad (19)$$

the value of 0 or 1 where each bᵢ is having.
As depicted in Figure 1, in a size of 4 × 4 state matrix, arranged the AES 128-bit input block. By the use of $b_{i,j}$variable represented the matrix elements where j ≤ 3, 0 ≤ i and the number of row and column is denoted as i, j, respectively. For AES rounds are permitted based on bits size in keys variables. The concept of 256-bit key size is utilized in our experimentation and hence, 14 is the number of rounds utilized, by the use Nr rounds is represented. To every round to supply keys in AES, also utilized the algorithm of key scheduling. By the transforms of various rounds, process the matrix of the input state. Via the cipher's various steps, as it passes and evolve the state matrix, and the ciphertext is produced finally. The following steps are followed by every round in AES.
SubBytes: In the AES, nonlinear step is T. To the state matrix bytes, the applied S-box is utilized by it. By their multiplicative inverse replaces every bite of the state matrixes, accompanied using a fine-mapping as follows:

$$b_i' = b_i \oplus b_{(i+4)mod8} \oplus b_{(i+5)mod8} \oplus b_{(i+6)mod8} \oplus b_{(i+7)mod8} \oplus c_i, for\ 0 \le i < 8 \qquad (20)$$

Where the i^th^ bit of the byte is b_i and the byte c's i^th^ bit is c_i with the value 01100011 or 63. So that, by the relation y = A.x-1 + B, to the S-box's output y, related is the input byte x, where constant matrices [27] are A and B.
Sift Rows: Through a byte position of a certain number, the state matrix's last three rows are rotated. It is carried out as follows:

$$+Nb))modNb) for\ 0 < r < 4and0 < c < Nb \qquad (21)$$

In the state matrix, the number of words is the Nb (in the state matrix, as a word, each column is considered). 128 bits are the input size as always Nb = 4 in AES and in size 4 x 4 of state matrix it arranged. As s in the state matrix every cell is represented with the index of column c and row r.
Mix columns: Column-by-column basis the state matrix operated this transformation, over GF (2^8), as polynomials of four-term, every column is considered and with a fixed polynomial of x and a modulo x^{4+1} is multiplied, given by

$$01\}x^1 + \{02\}$$

(22)
With the state matrix's columns, the multiplication process is given by

$$s'(x) = a(x) \otimes s(x)$$

Where multiplication modulo of polynomials is $a(x)$ and in the state matrix is s(x).

AddRoundKey: By an XOR operation of a simple bitwise, to the state added around the key in this process. From the key schedule, the size of Nb words is every round key is having. To fulfill the below circumstance, to the state matrix columns those Nb words are added:

$s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{\text{round} \times Nb+c}]$, for $0 \leq c < Nb$ (23)

where bitwise XOR is the ⊕ and the round number at the which round key is added is round and 0 ≤ round < Nr. Excluding the last round, the AES for each of the rounds performs all these steps.
The AES except the last round performs all these steps for each of the rounds. Not perform the Mix Column step in the last round. Figure 4 shows the process of round function for an AES of 14-round.

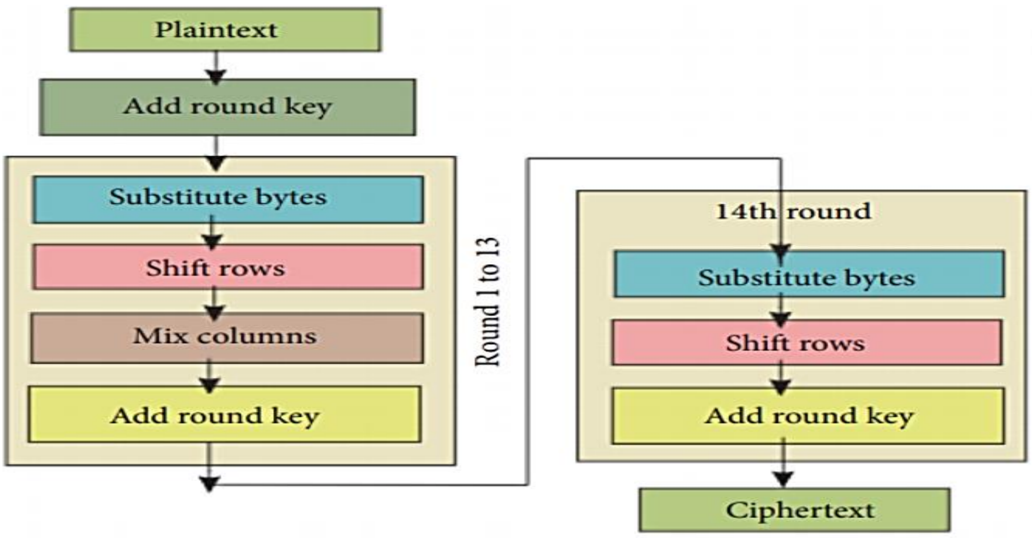


Figure.4: steps of round function in the AES of 14-round

In the stages of round function, one of the necessary parts is round keys adding, using a routine of key expansion generate these keys. A total of $Nb(Nr + 1)$ words are generated by the key expansion: an Nb word of the initial set is required by the algorithm, and the key data's Nb words are required by every Nr rounds. A 4-byte words linear array is contained in the resulting key schedule, denoted by [wi], 0 ≤ Nb≤ (Nr + 1). As input, these 4-byte words are taken by the Sub Word () function and then for every word, the S-box is applied. For a circular permutation to perform using another function Rotword (). In the below equation with x^{i-1} powers of x the values denoted as [x^{i-1} , {00}, {00}, {00}] are contained in the round constant array Rcon[i]:

$$\text{Rcon}[i] = x^{(i-4)/4} mod(x^8 + x^4 + x^3 + x + 1), wherei\ is\ the\ current\ round \quad (24)$$

Then for 192- and 128- bit keys there is a slight difference for a 256-bit key (Nk = 8) in the routine of key expansion. If i-4 is a multiple of Nk and Nk = 8, then before the XOR to w[i-1] apply the Sub Word (). A key's number of 32-bit words is Nk.

3.3.2 RK-AES

The AES algorithm's main problem in the expansion of key is that every word w_i is related to each other's. If traceable is any word, cryptanalysis linear methods or differential method by which we can find the overall key. To the algorithm, the confusion characteristics are provided as depicted in Figure 5, though the shifting function, S-boxes, and XOR operation, to the original key space can get back easily by the process of reverse engineering. To gain the key space partially, the words differences are revealed by the key space's biased inputs. With Symmetric Random Function Generator (SRFG) [7], the AES's module of key expansion is modified in AES to resolve this problem. Irrespective of the input string in the output string the sense of the number of 0's and 1's in which the symmetrically balanced output is produced by SRFG. A combined function output is its which consists of (XOR, OR, AND, and NOT) universal GATEs. for the generator of the proposed combined function, expression Te is provided as

(25)

Where i = 1, 2, 4, AND, XOR, OR, and NOT are the four universal GATES: the expression length is denoted by L (in the combined function f_c the number of terms); and the random combination is denoted by ⊗. L = 5 is used in our experiments. In the generator of such combined function the randomness is emphasized, in terms of variables of N input ' further expressed the above equation randomness in the selection, as proven in (2).

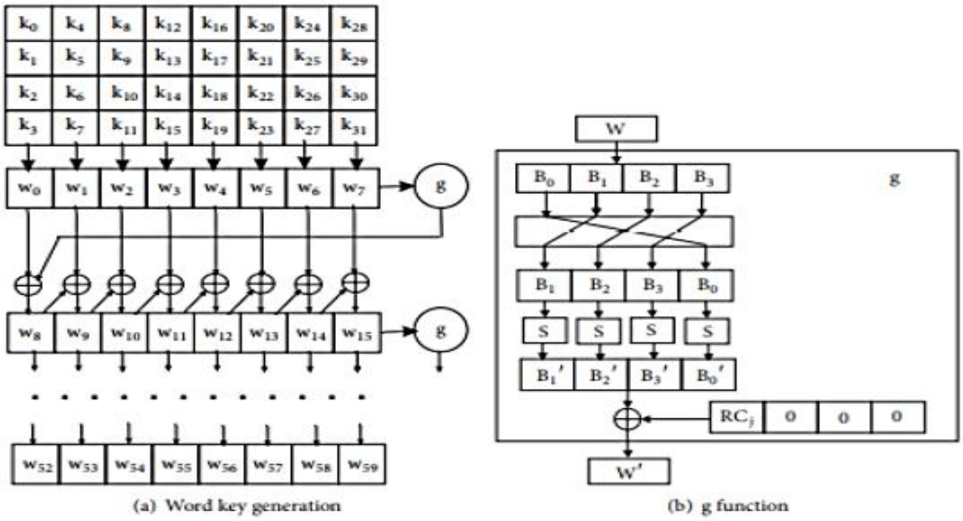
$_W)]$ (26)

The above equation is rewritten as below, for our experimentation,

(27)

With the feature of some randomness, the key expansion module is to be enabled in AES is the main objective of SRFG adding. Even though the partial key is in hand deducing the words of keys is prevented with the help of this. Figure 4 shows the module of modified key expansion. In yellow color highlighted the changes. In three parts the SRFG's randomness has been utilized: initially, in the g function, secondly, from key spaces the generation recursive word, and thirdly however most prominently, the addition of SRFG and RC from w_0 to w_7 for generating the words. As w, word every column in the key space is treated as per fig 6(a). in the very first step the eight worth w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7 we shall have, as 256 bits is the size of the key. Via a function g, w_7, is going which is the 8th word.

As in Figure 6, (b) just before the function's output SRFG is also utilizing by this function. Using a series of SRFGs processing, other words are generated by the use of the output of g. In AES for the 14 rounds until we get the necessary number of words repeat the same process. For the process of decryption, the generated words have been saved and to get back to the plaintext, with the ciphertext, reversely used them. For decryption rather than storing the keys, we shall work on the direct transmission of them in future.


k0 k4 k8 k12 k16 k20 k24 k28
k1 k5 k9 k13 k17 k21 k25 k29
k2 k6 k10 k14 k18 k22 k26 k30
k3 k7 k11 k15 k19 k23 k27 k31
w0 w1 w2 w3 w4 w5 w6 w7
g
w8 w9 w10 w11 w12 w13 w14 w15
w52 w53 w54 w55 w56 w57 w58 w59
W
B0 B1 B2 B3
B1 B2 B3 B0
S S S S
B1' B2' B3' B0'
RCj 0 0 0
W'

(a) Word key generation (b) g function

Figure.5: for AES of 14-round key expansion

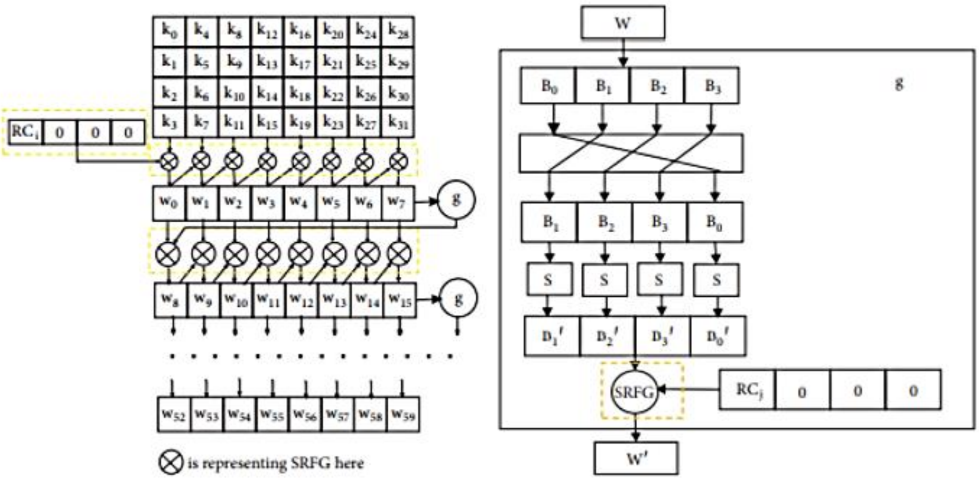

k0 k4 k8 k12 k16 k20 k24 k28
k1 k5 k9 k13 k17 k21 k25 k29
k2 k6 k10 k14 k18 k22 k26 k30
k3 k7 k11 k15 k19 k23 k27 k31
RCj 0 0 0
w0 w1 w2 w3 w4 w5 w6 w7
g
w8 w9 w10 w11 w12 w13 w14 w15
w52 w53 w54 w55 w56 w57 w58 w59
⊗ is representing SRFG here
W
B0 B1 B2 B3
B1 B2 B3 B0
S S S S
B1' B2' B3' B0'
SRFG
RCj 0 0 0
W'

Figure. 6 (a): generation of word key
Figure. 6 (b): for AES of 14 round proposed key expansion

3.3.3 Feature Analysis of RK-AES

AES-14 round's the module of key generation is emphasized, from the overall key byte’s deduction, can be removed the key bytes effect of biased inputs. From the module of key expansion out of the words generated a differential or linear equation can infer by the cryptanalysis process then the keys are deducing. To have the whole key in hand is not necessary always for the process of cryptanalysis. As we can see from the literature study, establishing such relations or deducing keys from subkeys is becoming faster and more complicated as cryptanalysis technology advance. Immunity, criterion, propagation, resiliency, balancedness, and nonlinearity are some parameters for RK-AES, for our proposed module of key expansion we have identified.

Every word w_i in the keyspace is consists of 2 bats (4 bytes) in our experimentation which is considered as a word vector of 32 -bit. Let S_2 be the set of all symmetric random combined functions on two variables of all the functions from F_2^2 into F_2 where $F_2^2 = (w_1, w_2) \mid w_i \in F_2 \cdot F_2$ is the finite field of two elements 0,1 and $\bigoplus$ is any operation of the field F_2.

As a polynomial, expressed any combined function $f_c \in B_2$ of five terms, which is function’s Algebraic Normal Form (ANF), termed as and provided as

$$f_c(w_1, w_2) = \bigoplus \lambda_u (\prod_{i=1}^{2} rand(w_i)^{u_i})^s \lambda_u \in F_2, u \in F_2^2 \; and \; L \in Z \qquad (28)$$

$$\bigoplus f_c(v), \;\; w \leq u, \forall w_i = w_{i_0}, w_{i,\ldots\ldots i_2} \qquad (29)$$

Where, $\left(w_{h_h}, w_{k_4} \ldots . w_{i_0}\right) \leq (u_1, u_2 \ldots, u_{32}) if V i, j, w_{i,} \leq u_i and j = 1,2, \ldots ,32$
(30)

On the input variables (number of i’s in the variable) weight, the output of f_c depends on. As a result, f_e corresponds toa function $g_c\colon \{0, .1, \ldots ,32\} \rightarrow F_2$ such that $\forall x \in F_2^2, f_e(x) = g_k(\omega t(x))$. As simplified value vector of f_c the sequence $g_\theta(f_t) = (g_t(0), g_\epsilon(1) \ldots \ldots g_e(32))$ for word vector of 32 -bit is considered. To establish the relationship between the arithmetic normal form and simplified value vector, (ii) can be rewritten as shown in (13).

$$d_f(j) \in \left(\prod_{i=1}^{2} rand(w_i)^{-j}\right)^2 = e\lambda_f(j) x_{jN} \qquad (31)$$

With 2 variables, where $\lambda_f(j), u \in F_2^2$, and $L \in Z, j = \{1,2\}, x_{jN}$ is the elementary polynomial of degree j. By 32 -bit vector represents the arithmetic normal form of f_ϵ‘s coefficients, $\lambda(f_c) = \{\lambda_f(0), \lambda_f(1), \ldots, \lambda_f, (32)\}$ called the ANF of f_c’s a simplified vector.

Nonlinearity: To prevent the various types of linear or related or even correlation attacks, in cryptographic algorithms the cryptographic functions shows an important design characteristic known as Nonlinearity. On the bits of the word vectors w_i this feature is depending on, and is considered as affine transformations of the function generated from the SRFG used. Between two affine transformations by the hamming distance, the nonlinearity is calculated, two words are w_j and w $w_i$of 32 bits each.

$$N1\left(w_{ik}, w_{jk}\right) = \sum_{k=1}^{n} w_{i_k} \neq w_h, wheren = 32 \qquad (32)$$

As subkeys, 4 words (128 bits) are used in AES in each of the rounds. Between the subkeys of two, the nonlinearity used for any two rounds r_i, r_j can be calculated as

$$\sum_{i=1}^{n} r_{ia} \neq r_{jk^*} \; where \; n = 128 \qquad (33)$$

Balancedness: If the following condition is followed by our key expansion of a proposed function f_e’s simplified value vector g then exists the balanced property:

$$\forall i = (1,2) g_c(t) = g_k(2 - i) \bigoplus = 1 \qquad (34)$$

where$\bigoplus$ is sum over F_2

To the symmetric functions correspondingly, the feature of trivial balancedness is also provided by the above equation. So that, the condition $D_1 f_c = 1$ is verified by f_c. For the even values of n (here for rounds, $n = 128$ and for words, $n = 32$), do not exist the functions having $D_1 f_c = 1$ because for any word vector w such that $\omega t(w) = n/2$ (where $\omega t(w)$ is the word vector's weight as the number of 1s in it is defined), the $D_1 f_c$ can be calculated as

$$(w) = f_i(w+1) = g\left(\frac{n}{2}\right) \equiv g_t\left(\frac{n}{2}\right) = 0 \qquad (35)$$

Resiliency: To the correlation attack [28] the correlation between the key expansion function output and its input variables small subset may leads cryptanalysis [29] of differential linearity. Hence, achieving the property of high resiliency becomes necessary for the function of key expansion. When remaining (n-m) bits are altered and any m input variables are fixed, if it remains balanced, then m -resilient is the N variable's function f_c, each of having n bits. if m is higher then, more resilient is the function. To some subspaces, to the f_c's restrictions weights, the resilient property is related. $V f_c \in B_2$and any affine any subspace $\delta\ C\ F_2^2$, the restriction of f_c to δ is the function given as

$$f_s\colon s \longrightarrow F_2 x \to f_c(x), \forall x \in \delta \qquad (36)$$

wherewith a $dim(\delta)$ variables function can be determined the f_s. By the k canonical bads vectors spanned the subspace δ and δ¯ is its supplementary subspace. By at δ where $a \in S$, the restrictions of f_c to ε all its costs and ε are given. Being f_ϵ balanced and symmetric, δ is represented as $\delta = (s_1, s_2 \dots, s_k)$and f_{ats} becomes balanced and symmetric toa. Moreover, for all $s \in \delta$, we can write the following:

$$f_{\mathrm{ats}}(s) = f(a+s) = g_\varepsilon(wt(a) + \mathrm{wt}(s)) \qquad (37)$$

when$a$ is fixed, upon the weight of s which depends. From f_c can be deduced these, the simplified ANF vector and simplified value vector as given below.

$$g_{l_{k+t}}(i) = g_k(i + wt(a)), \forall i, 0 \leq i \leq k\lambda_{t_{fe1}}(i) =* \lambda_f(i+j) \qquad (38)$$

$$\lambda_{t_{fe1}}(i) =* \lambda_f(i+j)$$

$$\forall_i, 0 \leq t \leq k\text{and}j \leq \mathrm{wt}(a)$$

Propagation Criterion: The cryptographic prompts of function derivatives determine the propagation criterion. For a cryptographic function's efficiency. To all its derivatives, its properties need to be propagated by the function. When the fixed Hamming weight of n/2 [7] they have a linearly equivalency in the key expansions of all derivatives. Sashes the propagation criterion of order m and degree k, from our previous work [7] n variables are applied in our key expansion proposed approach and by keeping m input bits constant, from the outputs obtain any affine function.

The propagation criterion of degree k is satisfied by Communication and Security Networks. For experimentation Considering every round, the following has done. Let $f_\epsilon \in B_2$ and let $r_i \cdot r_j \in F_2^2, \forall i, j = 1,2, \dots, 14$

Such that $t\ (r_i) = wt(r_j) = n/2$. Then, $D_{r_i} f_c$ and $D_{r_i} f_c$ are linearly equivalent.

This signifies, with a linear permutation μ of $F_2^2, D_{r_i} f_c = D_{r_i} f_c \cdot \mu$ if we change the input variables, where the composite function is o. On the variable the permutation μ exists in a way so that that $r_j = \mu(r_i)$. Since balanced and symmetric is f_d, we can have

$$D_{r_j} f_e(\mu(a)) = D_n f_c(a),\ where\ a \in \delta^- \qquad (39)$$

and $e_k = w_{n-k+1} + \dots + w_n$. Then for any $z = a + r_j$, with $a \in \delta^-$. then we can have the following:

$$wt(z) = \omega t(a) + wt(r_j)\omega t(z + \epsilon_k) = wt(a) + wt(r_j + \epsilon_k) = wt(a) + k - \omega t(r_j)$$

(40)

Thus, $\forall a \in V$.

$$D_e f_c(a+y) = f_c(a+b)\, w\, f_c\left(a+\epsilon_k+r_j\right) = wt(a) + wt\left(r_j+\epsilon_k\right) = wt(a) + k - \; ut\left(r_j\right) = g_e\left(wt(a)+w\left(r_j\right)\right) \equiv \left(\omega t(a) + k - w\left(r_j\right)\right) \tag{41}$$

The symmetric property is followed by g_ε which is signified by the equation (28). With the propagation features, also propagate our key expansion of proposed output's partial derivatives.

Immunity: With the 32 bits (on bit size has been done no modification) variables (words) the module of proposed key expansion is performed. In concern algebraic immunity and correlation immunity are two types of immunity and as a binary vector of 32 -bit, among the two input variables $w_j$is considered for correlation immunity, the correlation immune is the output if

$$Prob(f_c = w_i) = \frac{1}{2}, 1 \leq i \leq 32 \tag{42}$$

All the bits must be equal to the probability distribution and for that the output words w_e has the following property:

$$|\mathrm{m}[M_0(w_\infty(w_a)^r) - M_1(w_\infty(w_e)^r)]| = \mathrm{m}[m] \to 0 \tag{43}$$

where$[M_0(w_e,(w_e)^r)]$ is the reverse of the matching of output words from the key expansion process with respect to value 0, and $[M_1(w_{e^+}(w_e)')]$ is the reverse of the matching of output words from the key expansion process with respect to value 1. Following the aforementioned property, an intriguing aspect of our proposed key expansion module has been recognized, and the following proposition has been given.

Proposition 1. In AES-256, if $\left[M_0\left(w_{b_2}(w_0)^r\right)\right] = m_0$and $[M_1(w_b,(w_e)^r)] = m_1$, then $Nl(w_e,(w_b)^r) = m_0 + m_1$.
To the annihilator of a function [30], algebraic immunity is related. The following can consider for our proposed key expansion to evaluate this property.

Given, $f_\varepsilon \in B_2$, as the function $f_c's$ annihilator define any function of the set $A(f_e) = \{g \in B_2 \mid gf = 0\}$. All non-zero annihilators of $f(c)+1$or $f(c)$'s minimum degree is $AI(f_5)$ which is used to denote the algebraic immunity of f_s. The $AI(f_e)'s$ value is given as

$$AI(f_c) = \; m[deg(g) \mid g \neq 0, g \in A(f_a) \cup A(f_c+1) \tag{44}$$

As to generate the output words we have used SRFG, always $n/2$ is the minimum degree. So that, always $n/2$ is the output's algebraic immunity from its, which is optimal always.

3.3.4 Security Analysis of RK-AES

By the use of SRF, in AES-256 the modification of proposed key expansion overall features is analyzed in the above section. On our key expansion module of modified AES, the security analysis is performed in this section to justify the features. Fault analysis attacks and related attacks are the two attacks we have considered.
Related Key Attack Analysis: To deduce the original key, among the keys differential relations or linear relations are used in Related key attacks.
For input, nonzero word difference be an*nz* and for the input difference *nz*, S-box's output difference isan0. With these differences to execute the attack, one of $2^{14}-1$ values can be the difference 0, because of the operation of XOR's symmetry utilized in the algorithm of generic AES-256 and including the whitening of keys one of $2^{15}-1$ differences can be *nz* difference. Also, higher is the key to deducing probability, when in a bounded value region these differences are. This difference is increased by the feature of nonlinearity in our proposed modified AES and hence, also drastically increases the key space of searching. The below formula increased the searching space complexity in key space for a word of 32-bit:
For key space search's complexity = $2^{32}.2^{Nl}$
In the AES's proposed key expansion, the value of nonlinearity where *Nl* and *Nl* = 20.7 is the average value. Therefore, on AES greater than the complexities of key searching of differential attacks, this

complexity becomes $2^{52.7}$. In differential attacks our proposed algorithm is preventive which is shown by this.
Moreover, as $K_{u1}, K_{u2}, K_{u3}, K_{u4}$ are the unknown keys but four related keys used by the attacker. To recover K_{u1} is the attacker's objective. To establish the attack the required relation is

$$K_{u2} = K_{wt} \oplus \varDelta K^{*} \quad (45)$$

$$K_{u3} = K_{w} \oplus \triangle K' \quad (46)$$

$$K_{u4} = K_{ut} \oplus \triangle K^{*} \oplus \triangle K' \quad (47)$$

For 1 to 7 rounds, to the D° which is a first related-key differential uses the difference of cipher key $\triangle K^{*}$ and for 8 to 14 rounds, for the second related-key differential $D^{\dagger}$ uses the cipher key difference $\triangle \underline{K'}$. Assuming that information regarding $\triangle K'$ and $\triangle K^{*}$ only the attacker has. To recover any 32 -bit words (any word out of the 60 words) the back-tracing probability is calculated as

$$P(w_i) = \frac{1}{(2^{n}.i.P_{L}^{L}.2^{V})} \quad (48)$$

For our modified proposed key expansion of AFS-256, $n = 32$ is the number of bits in each word, $i = 60$ is including the whitening keywords, the total number of words, $L = 5$ is the number of expression length totally, and $V = 2$ is for each operation, the total number variables utilized. By the use of the values, the probability becomes as below

$$P(w_i) = \frac{1}{(2^{32} \times 60 \times P_{1}^{0} \times 2^{2})} = \frac{1}{2^{3} \times 225} \quad (49)$$

By the use of our proposed key expansion approach, to recover an AFS-256's single word, too less is the probability which is shown by the above result.

Instead of using a simple XOR operation, by the use of SRFG generate the words which are shown in figure 4. So that, using SRFG for our proposed solution of AFS, (44), (45), and (46) will not be feasible. It means the related attack resistance is the solution proposed. Moreover, in the key deducing $\triangle K'$ and $\triangle K^{*}$ are a factor. However, as high nonlinearity is provided by our proposed solution, to recover the key space words, $\triangle K'$ and $\triangle K^{*}$ are not suitable. From our experimentation's observation, a proposition is inferred as follows.

Proposition 2. For the first related-key differential D° the difference of cipher key $\triangle K^{*}$ is used and the cipher key difference $\triangle K'$ is used for the second related-key differential are considered D^{1}, to the non-linearity inversely proportional is the nonlinearity.

$$D^{0} + \varDelta K^{*} \propto \frac{1}{Nl} and\, D^{1} + \triangle K' \propto \frac{1}{NI} \therefore D^{\circ}. D^{3} + \triangle K^{*}. \triangle K' \propto \frac{1}{NF}$$
(50)

Fault Injection Analysis: In the bytes of key only the fault injection is considered in this part. For the original key byte of any random, in the key matrix injected the faulty key byte is assumed. From all 1 bit's bytes or all 0 bits, the byte's biased input inferred the faulty input. The relationship among words of round or even word byte is revealed by the use of such based inputs and faulty in the original AES. Hence, as in the literature review, less complexity reduced the key recovery space in the original AES. For AES-256 the following proposition we can have by recollecting (12).

Proposition 3. With expression terms and variables of two, using SRFG for AES-256, with any two random faulty bytes the key recovery complexity is calculated as

$$\text{Prob}(FI) = \frac{1}{\Sigma \oplus \lambda_{u} \left(\Pi_{i-1}^{2} \text{rand}(w_i)^{\mu}\right)' C_{2}^{\infty}} \quad (51)$$

For faulty key byte of any random, balanced, and nonlinear is always the layered SRFGs output. Therefore, to other bytes, the fault is not further propagated as the linear equations and/or the differences became invalid according to Proposition 2. Hence, even in the fault injection bytes preventive is our proposed key expansion. Algorithm 1 shows the RK-AES Algorithm.

```
Algorithm 1.     RK-AES Algorithm

Step 1: for input, nonzero word difference be annz and for the input difference nz the S-box's output difference be an o.

Step 2: In the keyspace for a word of 32-bit, with the following formula the searching space complexity increases:
        Complexity for key space search = $2^{32}.2^{Nl}$

Step 3: To establish the attack relation required is set as

        $Ku2=Kwt\oplus\Delta K^*$      $Ku3=Kw\oplus\Delta K'$          $Ku4=Kut\oplus\Delta K^*\oplus\Delta K'$

Step 4: If round (1 to 7) {
        for the first related-key differential D the difference of cipher key used is the K*
}
Else if round (8 to 14) {
K' is the cipher key difference used for the second related-key differential D
}

Step 5: calculate the back-tracing probability
$\mathrm{P(w_i)} = \frac{\mathbf{1}}{\left(\boldsymbol{2^{n}i\cdot P_{L}^{L}}\right)^{\boldsymbol{v}}}$
```

IV. PERFORMANCE ANALYSIS

This model is implemented over python programming language in which the system specification used for building are Windows 10 OS, 10th Gen Intel Core i7, Nvidia RTX 3080 Max-Q GPU. Here we evaluated our proposed model (AES-RK-DBN) with other models such as LSTM, DenseNet, Resnet, VGG, CNN underperformance measures like Accuracy, specificity, sensitivity, detection rate, Recall, F-score, False Positive Rate (FPR), True Positive Rate (TPR), Security, encryption time, memory utilization, throughput encryption time. Table 2 depicts the comparative analysis of various models with our proposed model under 3 datasets. Figure 7 (a, b, c) depict the graphical representation of the various model concerning our model underperformance measure like sensitivity, specificity, accuracy over KDD-cup, NSL-KDD, UNSW-NB15 dataset.

Table 2: comparison analysis

Models	Dataset	Sensitivity (%)	Specificity (%)	Accuracy (%)

VGG LSTM DenseNet CNN AES-RK-DBN	KDD-Cup 1999 [62]	88 87.3 89.6 90.67 91.2	78 84 88 81 89	85.9 87.4 90.3 93.1 94.4
VGG LSTM DenseNet CNN AES-RK-DBN	NSL-KDD [54]	83.9 79.5 83 85 84	82 85 88.9 80.3 85	80 86.2 82.8 87 95
VGG LSTM DenseNet CNN AES-RK-DBN	UNSW-NB15 [49]	83.6 82.9 86.6 87.8 88.4	81 84.5 86.3 83.9 88.3	82.9 86.5 84.5 89.7 94

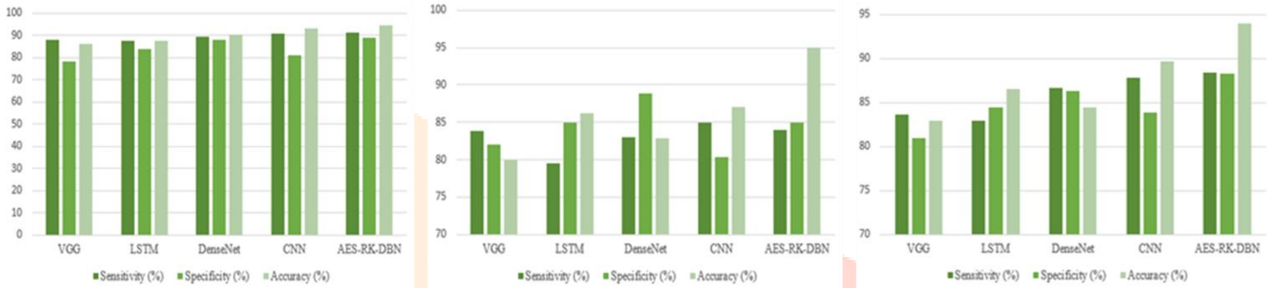


Figure. 7: (a, b, c) model vs sensitivity, specificity, accuracy over KDD-CUP dataset Vs NSL-KDD dataset Vs UNSW-NB15 dataset

Table 3. depicts the average accuracy, specificity, and sensitivity of various models for predicting the normal/ malicious data. Figure 8 depicts the graphical representation of average performance measures of all models with our proposed model (AES-RK-DBN) in which our model shows better accuracy 94.1% when compared to other models.

Table 3: average performance measures [49]

Models	**Sensitivity**	**Specificity**	**Accuracy**
VGG	83.6	81	82.9
LSTM	82.9	84.5	86.5
DenseNet	86.6	86.3	84.5
CNN	87.8	83.9	89.7
AES-RK-DBN	88.4	88.3	94

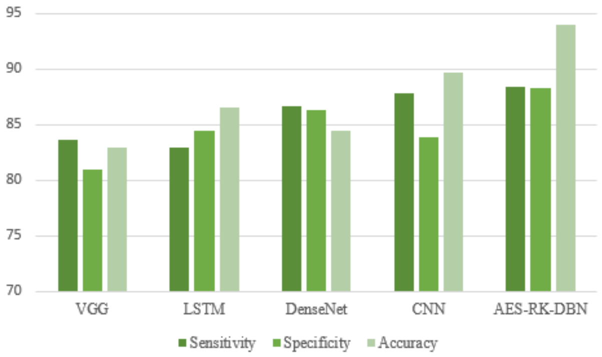


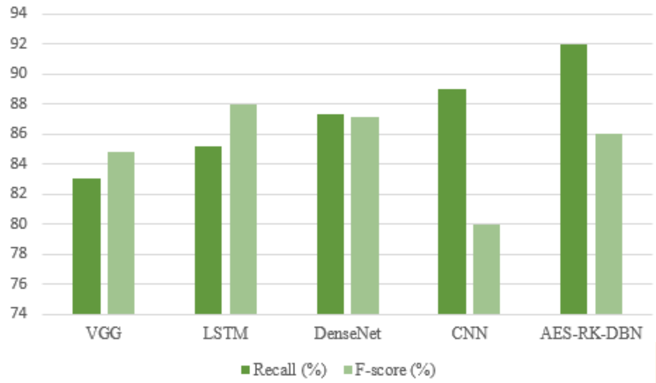
Figure.8: models vs average (sensitivity, specificity, accuracy)

Table 4. depict the performance analysis of various models under detection rate, recall, f-score. Figure 9 depicts the graphical representation of recall and f-score measures of various model's vs our models. Figure

10 depicts the detection rate of various models in which our model has a good detection rate of 0.94 when compared to other models due to the RBM stack layers.

Table 4: recall and f-score of various models [39]

Models	Recall (%)	F-score (%)
VGG	83	84.8
LSTM	85.2	88
DenseNet	87.3	87.1
CNN	89	80
AES-RK-DBN	92	86



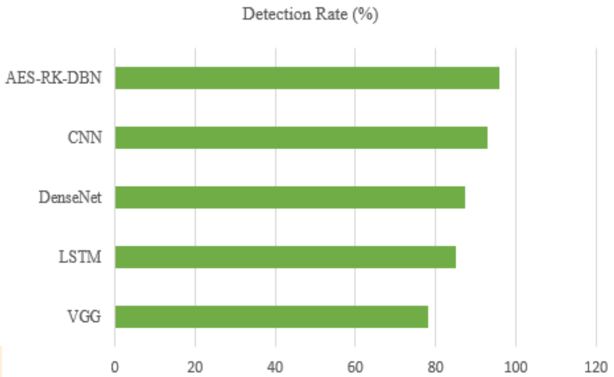


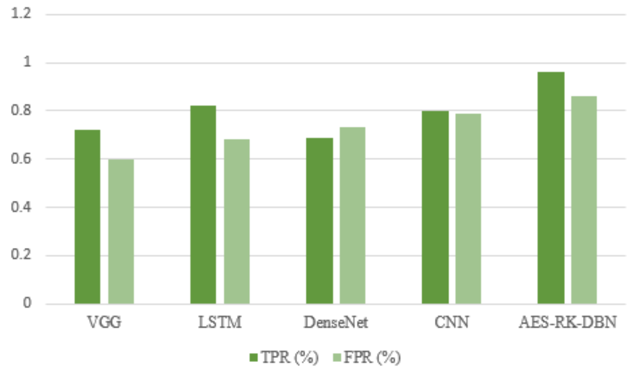
Figure.9: models’ vs recall, f-score

Figure.10: models’ vs detection rate

Table 5 depicts the TPR and FPR of various models and depicting how much they classify normal and malicious data. Figure 11 shows a graphical representation of various models concerning our model in which our proposed model classifies normal at 0.96 rates and Figure 12 shows a graphical representation of various models concerning our model in which our proposed model classifies malicious as 0.8 rates when compared to other models.

Table.5: TPR and FPR of various models [27] [34]

Models	Classification	TPR (%)	FPR (%)
VGG LSTM DenseNet CNN AES-RK-DBN	Normal Data	0.72 0.82 0.69 0.8 0.96	0.6 0.68 0.73 0.79 0.86
VGG LSTM DenseNet CNN AES-RK-DBN	Malicious Data	0.8 0.78 0.83 0.79 0.8	0.62 0.72 0.79 0.83 0.81



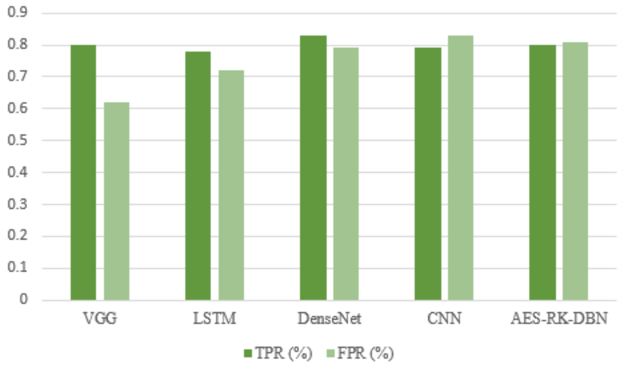


Figure.11: TPR and FPR of normal data Figure.12: TPR and FPR of malicious data

Table 6 depicts the security, encryption time, memory utilization, throughput encryption time of various models based on encryption in which our model outperforms better than any other model. Figure 13 depicts the graphical representation of various models under encryption time, throughput encryption time. Figure 14 depict the graphical representation of various model under memory utilization measure. Figure 15 depicts the graphical representation of various models under security in which our model with help of AES, has high-level security compared to other models. Figure 16 depicts the comparative analysis of our proposed model with state-of-art models.

Table 6: performance analysis based on encryption [22] [24]

Models	Encryption Time (s)	Memory utilization (%)	Throughput Encryption Time (s)	Security (%)
VGG	0.78	91	0.5	70
LSTM	0.6	93	0.7	80
DenseNet	0.5	95	0.5	75
CNN	0.7	90	0.8	85
AES-RK-DBN	0.4	89	0.9	95

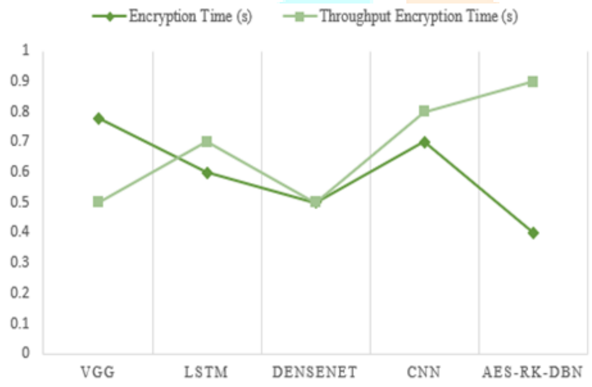


Figure 13: models vs encryption time and throughput encryption time

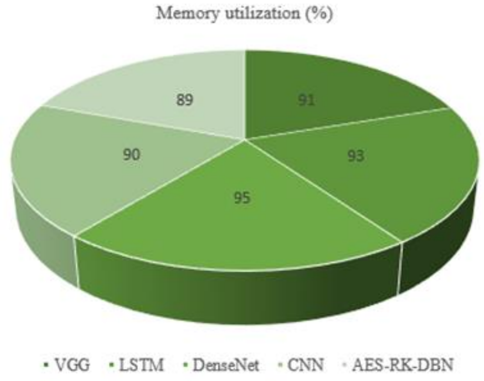


Figure 14: models vs memory utilization

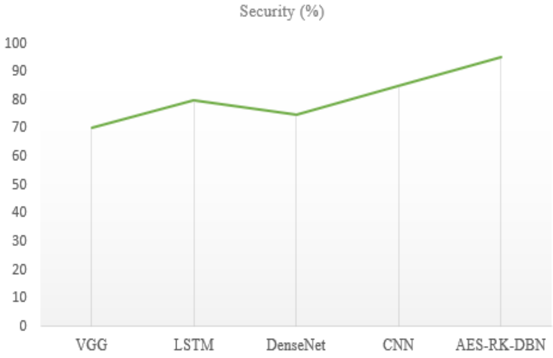


Figure 15: models' vs security

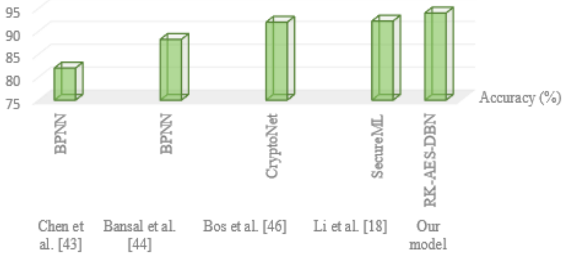


Figure 16: the comparative analysis proposed model with state-of-art models

V. CONCLUSION

In allusion to the shortcomings of traditional local computing technology, cloud computing is an emerging technology that is used by millions of people for various purposes. The security of cloud data from malicious threats is an important task that can be effectively achieved by our proposed AES-RK-DBN Model. The proposed model is the combination of the Deep Learning Model by DBN and an Encryption Model by the RK-AES Algorithm. This makes this a more secure and accurate cloud computing model. The proposed model is further compared with the existing models such as LSTM, DenseNet, and VGG under various measures in which our model gives a much secure level, 94.1 % of accuracy in detecting threats, 0.95 detection rate, and greater encryption time.

REFERENCES

1. Ellen Messmer (2012). Gartner: Growth in Cloud Computing to shape 2013 security trends, Network World [Online]. Available: http://www.networkworld.com/news/2012/120612- Gartner-cloud-security-264873.html
2. SachdevAbhaThakral, and MohitBhansali. "Addressing the Cloud Computing Security Menace." IJRET, Volume 2, Issue 2, pp. 126-130, Feb 2013.
3. Chen, Yao, and RaduSion. "On securing untrusted clouds with cryptography. “Proceedings of the 9th annual ACM workshop on Privacy in the electronic society. ACM, 2010.
4. Talbot, David (2009). “How Secure Is Cloud Computing?” Technology Review [Online]. Available: http://www.technologyreview.com/computing/23951/
5. Agudo, Isaac and Nuez, David and Giammatteo, Gabriele and Rizomiliotis, Panagiotis and Lambrinoudakis, Costas. Cryptography Goes to the Cloud. In Lee, Changhoon and Seigneur, Jean-Marc and Park, James J. and Wagner, Roland R., editors, Secure and Trust Computing, Data Management, and Applications, pages 190–197, Springer Berlin Heidelberg, 2011.
6. Op-ed: Encryption, not restriction, is the key to safe cloud computing. Available Online: http://www.nextgov.com/cloud-computing/2012/10/oped-encryption-not-restriction-key-safe-cloudcomputing/58608/
7. “Cloud Security and Privacy”, Tim Mather, SubraKumaraswamy, and ShahedLatif – O’Reilly Book.
8. Elminaam, DiaaSalama Abdul, Hatem Mohamed Abdul Kader, and Mohie Mohamed Hadhoud. "Performance Evaluation of Symmetric Encryption Algorithms." IJCSNS International Journal of Computer Science and Network Security 8.12 (2008): 280-286.
9. Sanchez-Avila, C., and R. Sanchez-Reillol. "The Rijndael block cipher (AES proposal): a comparison with DES." Security Technology, 2001 IEEE 35th International Carnahan Conference on. IEEE, 2001.
10. P. Mohassel and Y. Zhang, ‘‘SecureML: A system for scalable privacypreserving machine learning,’’ in Proc. IEEE Symp. Secur. Privacy, May 2017, pp. 19–38
11. NIST, FIPS PUB 197, “Advanced Encryption Standard (AES),” November 2001 [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips197.pdf.
12. D. T. Hoang, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: Architecture, applications, and approaches,” Wireless Communications and Mobile Computing, vol. 13, no. 18, pp.1587-1611, Dec. 2013.
13. Louis Columbus, Roundup of cloud computing forecasts and market estimates 2016, Forbes magazine
14. 2015 information security breaches survey, Technical Report, PWC.
15. J. Cao, B. Yu, F. Dong, X. Zhu, and S. Xu, “Entropy-based denial-ofservice attack detection in cloud data center,” Concurrency and Computation: Practice and Experience, vol. 27, no. 18, pp. 5623-5639, Dec. 2015.
16. M. N. Ismail, A. Aborujilah, S. Musa, and A. Shahzad, “Detecting flooding-based DoS attack in cloud computing environment using covariance matrix approach,” in IEEE ICUIMC, Kota Kinabalu, Malaysia, Jan. 2013.
17. A. Sahi, D. Lai, Y. Li, and M. Diykh, “An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment,” IEEE Access, vol. 5, pp. 6036-6048, Apr. 2017.
18. M. Chouhan and H. Hasbullah, “Adaptive detection technique for cachebased side channel attack using Bloom Filter for secure cloud,” in IEEE International Conference on Computer and Information Sciences, pp. 293- 297, Aug. 2016.
19. P. Li et al., ‘‘Multi-key privacy-preserving deep learning in cloud computing,’’ Future Gener. Comput. Syst., vol. 74, pp. 76–85, Sep. 2017.
20. K. Wang and Y. Hou, “Detection method of SQL injection attack in cloud computing environment,” in IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, pp. 487- 493, Oct 2016.
21. B. C. Youssef, M. Nada, B. Elmehdi, and R. Boubker, “Intrusion detection in cloud computing based attacks patterns and risk assessment,” in International Conference on Systems of Collaboration, pp. 1-4, Nov. 2016.
22. A. Nezarat, “A game theoretic method for VM-to-hypervisor attacks detection in cloud environment,” in Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 1127-1132, May 2017.
23. G. Nenvani and H. Gupta, “A survey on attack detection on cloud using supervised learning techniques,” in IEEE Symposium on Colossal Data Analysis and Networking, pp. 1-5, Mar. 2016.

24. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436-444, May 2015.
25. K. Kurihara and K. Katagishi, “A Simple Detection Method for DoS Attacks based on IP Packets Entropy values,” IEEE Asia Joint Conference on Information Security, Wuhan, China, Sept. 2014.
26. I. T. Jolliffe, Principal Component Analysis and Factor Analysis, Principal component analysis. Springer New York, 1986.
27. J. Shlens, “A tutorial on principal component analysis,” arXiv preprint arXiv:1404.1100, 2014. [16] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [17] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1-6, 2009.
28. G. E. Hinton, S. Osindero, and Y-W. Teh, “A fast learning algorithm for deep belief nets,” Neural computation, vol. 18, no.7, pp. 1527-1554, 2006.
29. G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786, pp. 504-507, 2006.
30. Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layerwise training of deep networks,” Advances in neural information processing systems, pp. 153-160, 2007.
31. G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural computation, vol.14, no. 8, pp. 1771-1800, 2002.
32. G. E. Hinton, “A practical guide to training restricted Boltzmann machines,” Momentum 9, no. 1, pp. 926, 2010. [23] N. Mowla, I. Doh, and K. Chae, “Evolving neural network intrusion detection system for MCPS,” in IEEE International Conference on Advanced Communication Technology, pp. 183-187, Feb. 2017
33. Computing to shape 2013 security trends, Network World [Online]. Available: http://www.networkworld.com/news/2012/120612- gartner-cloud-security-264873.html
34. SachdevAbhaThakral, and MohitBhansali. "Addressing the Cloud Computing Security Menace." IJRET, Volume 2, Issue 2, pp. 126-130, Feb 2013.
35. Chen, Yao, and RaduSion. "On securing untrusted clouds with cryptography. “Proceedings of the 9th annual ACM workshop on Privacy in the electronic society. ACM, 2010.
36. V. Chang, Y.-H. Kuo, and M. Ramachandran, ‘‘Cloud computing adoption framework: A security framework for business clouds,’’ Futur. Gener. Comput. Syst., vol. 57, pp. 24–41, Apr. 2016.
37. M. O. Alassafi, A. Alharthi, R. J. Walters, and G. B. Wills, ‘‘A framework for critical security factors that influence the decision of cloud adoption by Saudi government agencies,’’ Telematics Inform., vol. 34, no. 7, pp. 996–1010, 2017.
38. V. Chang and M. Ramachandran, ‘‘Towards achieving data security with the cloud computing adoption framework,’’ IEEE Trans. Services Comput., vol. 9, no. 1, pp. 138–151, Jan. 2016.
39. J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, ‘‘Secure deduplication with efficient and reliable convergent key management,’’ IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 6, pp. 1615–1625, Jun. 2014.
40. J. Li, Y. K. Li, X. Chen, P. P. C. Lee, and W. Lou, ‘‘A hybrid cloud approach for secure authorized deduplication,’’ IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 5, pp. 1206–1216, 2015.
41. D. Talia, P. Trunfio, and F. Marozzo, Data Analysis in the Cloud: Models, Techniques and Applications, vol. 1. Amsterdam, The Netherlands: Elsevier, 2016, pp. 1–150.
42. X. Ma, F. Zhang, X. Chen, and J. Shen, ‘‘Privacy preserving multi-party computation delegation for deep learning in cloud computing,’’ Inf. Sci., vol. 459, pp. 103–116, Aug. 2018.
43. T. Chen and S. Zhong, ‘‘Privacy-preserving backpropagation neural network learning,’’ IEEE Trans. Neural Netw., vol. 20, no. 10, pp. 1554–1564, Oct. 2009.
44. A. Bansal, T. Chen, and S. Zhong, ‘‘Privacy preserving back-propagation neural network learning over arbitrarily partitioned data,’’ Neural Comput. Appl., vol. 20, no. 1, pp. 143–150, 2011.
45. J. Yuan and S. Yu, ‘‘Privacy preserving back-propagation neural network learning made practical with cloud computing,’’ IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. 212–224, Jan. 2014.
46. J. J. W. Bos, K. K. Lauter, J. Loftus, and M. Naehrig, ‘‘Improved security for a ring-based fully homomorphic encryption scheme,’’ Cryptography and Coding, vol. 8308. New York, NY, USA: Springer, 2013, pp. 45–64.
47. M. Hesamifard, E. Takabi, H. Ghasemi, E. Hesamifard, H. Takabi, and M. Ghasemi. (2017). ‘‘CryptoDL: Deep neural networks over encrypted data.’’ [Online]. Available: https://arxiv.org/abs/1711.05189
48. H. Chabanne, A. de Wargny, J. Milgram, C. Morel, and E. Prouff, ‘‘Privacypreserving classification on deep neural network,’’ IACR Cryptol. ePrint Arch., Tech. Rep., 2017.

49. B. D. Rouhani, M. S. Riazi, and F. Koushanfar, ‘‘DeepSecure: Scalable provably-secure deep learning,’’ in Proc. 55th Annu. Design Autom. Conf. (DAC), San Francisco, CA, USA, Jun. 2018, pp. 2:1–2:6.
50. W. S. Hong, A. D. Haimovich, and R. A. Taylor, ‘‘Predicting hospital admission at emergency department triage using machine learning,’’ PLoS ONE, vol. 13, no. 7, 2018, Art. no. e0201016.
51. S. Ioffe and C. Szegedy, ‘‘Batch normalization: Accelerating deep network training by reducing internal covariate shift,’’ in Proc. 32nd Int. Conf. Mach. Learn. (ICML), Lille, France, Jul. 2015, pp. 448–456.
52. Talbot, David (2009). “How Secure Is Cloud Computing?” Technology Review [Online]. Available: http://www.technologyreview.com/computing/23951/
53. Agudo, Isaac and Nuez, David and Giammatteo, Gabriele and Rizomiliotis, Panagiotis and Lambrinoudakis, Costas. Cryptography Goes to the Cloud. In Lee, Changhoon and Seigneur, Jean-Marc and Park, James J. and Wagner, Roland R., editors, Secure and Trust Computing, Data Management, and Applications, pages 190–197, Springer Berlin Heidelberg, 2011.
54. Op-ed: Encryption, not restriction, is the key to safe cloud computing. Available Online: http://www.nextgov.com/cloud-computing/2012/10/oped-encryption-not-restriction-key-safe-cloudcomputing/58608/
55. “Cloud Security and Privacy”, Tim Mather, SubraKumaraswamy, and ShahedLatif – O’Reilly Book.
56. Elminaam, DiaaSalama Abdul, Hatem Mohamed Abdul Kader, and Mohie Mohamed Hadhoud. "Performance Evaluation of Symmetric Encryption Algorithms." IJCSNS International Journal of Computer Science and Network Security 8.12 (2008): 280-286.
57. Sanchez-Avila, C., and R. Sanchez-Reillol. "The Rijndael block cipher (AES proposal): a comparison with DES." Security Technology, 2001 IEEE 35th International Carnahan Conference on. IEEE, 2001.
58. NIST, FIPS PUB 197, “Advanced Encryption Standard (AES),” November 2001 [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips197.pdf.
59. Enterprise and Individual Users to fuel Growth in Cloud Computing [Online]. Available: http://www.redorbit.com/news/technology/1112692915/c loud-computing-growth-paas-saas-091212/
60. Worldwide and Regional Public IT Cloud Services 2012- 2016 Forecast [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=236552
61. John Harauz, Lori M. Kaufman and Bruce Potter, —Data security in the world of cloud computing —, 2009 IEEE CO Published by the IEEE Computer and Reliability Societies.
62. Jensen, Meiko, et al. "On technical security issues in cloud computing." Cloud Computing, 2009. CLOUD'09. IEEE International Conference on. IEEE, 2009
63. W. Stallings, Cryptography and Network Security: Principles and Practices, Pearson, 2005.
64. S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, “Public key authentication and key agreement in iot devices with minimal airtime consumption,” IEEE Embedded Systems Letters, vol. 9, no. 1, pp. 1–4, 2017.
65. S. Raza, L. Seitz, D. Sitenkov, and G. Selander, “S3K: Scalable security with symmetric keys - DTLS key establishment for the internet of things,” IEEE Transactions on Automation Science and Engineering, vol. 13, no. 3, pp. 1270–1280, 2016.
66. T.W. Cusick and P. Stanica,Cryptographic Boolean functions and applications, Elsevier/Academic Press, 2017.
67. J. Zhang, Y. Lin, Y. Lyu, and G. Qu, “A PUF-FSM Binding Scheme for FPGA IP Protection and Pay-Per-Device Licensing,” IEEE Transactions on Information Forensics and Security, vol. 10, no. 6, pp. 1137–1150, 2015.
68. J.-L. Zhang, G. Qu, Y.-Q. Lv, and Q. Zhou, “A survey on silicon PUFs and recent advances in ring oscillator PUFs,” Journal of Computer Science and Technology, vol. 29, no. 4, pp. 664–678, 2014.
69. R. Saha and G. Geetha, “Symmetric random function generator (SRFG): A novel cryptographic primitive for designing fast and robust algorithms,” Chaos, Solitons& Fractals, vol. 104, pp. 371– 377, 2017.
70. Q. Wang, A. Wang, L. Wu, and J. Zhang, “A new zero value attack combined fault sensitivity analysis on masked AES,” Microprocessors and Microsystems, vol. 45, pp. 355–362, 2016.
71. G. Piret and J. Quisquater, “A diferential fault attack technique against spn structures, with application to the AES and khazad,” in Cryptographic Hardware and Embedded Systems - CHES 2003, vol. 2779 of Lecture Notes in Computer Science, pp. 77–88, Springer, Berlin, Heidelberg, Germany, 2003.