Advancement of IoT Devices using LSTM for an Enhanced IoT Security

T.S.Sandeep

Senior Assistant Professor
Department of Information Technology
Sri Venkateswara Engineering College (SVEC), Tirupati, Andhra Pradesh
sandeep.t@svcolleges.edu.in

Abstract

Nowadays, the evolve of unknown attacks have been prevented with an advancement of cyber security whereas there are several applications like smart cities and smart industries have evolved through Internet of Things (IoT) which perform one of the developing fields. The major challenge faced in the IoTs infrastructure is due to incremental of cyber-attacks fashions. The digital world development into an environment of physical gets accumulated with recent area attacks over traditional interne with current security threat. However, the main challenges faced in the physical IoTs connection is about implementation of distributed security mechanism to IoT devices resource constrain. It is essential that IoT devices can be automatically monitored and upgraded as firmware which consists of vulnerabilities like buffer overflows and it needed to be patched. In order to receive a replacement firmware, the devices are allowed to connect with cloud server automatically has been processed through firmware update. Anomaly and malicious behavior detection are the crucial concern which has become a priority in the area of intrusion detection. At present, the methods of intrusion detection can be generally based on single point detection and maintaining which can't be able identifies the attacking mode with the frequency of hidden attack. The familiar detection technology is recently using conventional Machine Learning (ML) algorithms for training the sample of intrusion to accomplish the intrusion detection models but these algorithms have demerits of poor detection speeds. However, the deep learning is one of the advanced technologies which extract behaviors from samples automatically. Hence, the intrusion detection accuracy is not high over traditional ML technology. Thus, the study introduced a hybrid algorithm of Elliptic Curve Cryptography- Long Short Term Memory (ECC-LSTM) model in which ECC perform as an Edge Node which exists over a private network and is built to perform as a connecting network client for an online cloud server is comparatively secured than of devices that may perform as a server itself. Eventually, the LSTM based on anomaly graph tool to detect Collective contextual Anomalies (CCA) has been detected from the samples and even detecting the malicious behavior from the extracted behavior chains which get evaluated with its accuracy, False Negative Rate (FNR) and False Positive Rate (FPR).

Keywords: *LSTM, ECC, IoT, Transportation Layer Security (TLS), Anomaly, EDSA, Malicious Behavior Detection (MBD), CCA.*

1. Introduction

The role of IoT has become an essential factor in our daily lives, with the progressive advancement in IT industries. However, these inter-connective devices or sensors have the capability of collecting and transferring of various data between themselves via infrastructure of modern communication network associated with millions IoT nodes [1- 4]. Subsequently, there are several IoT application have provided

high accuracy and services of fine grained network to users. However, at present enormous amount of devices and sensors have been interconnected through platform of IoT and these devices and sensors are producing huge data and requirement for more processing that provide an intelligence for both users and service providers. All the data need to be uploaded in centralized server in traditional cloud computing and once the computation is over, the outcome have been send back to the devices and sensor. In addition, IoT has enabled the automated system for the environment like homes and industrial to enormous recent threats [5, 6]. There are various reasons to the IoT device security with sad state. The development team of IoT has frequently worked without adequate resources and under strict time limitations whereas these elements make it easy for engineering teams to cut corners. For an instance, re-using unverified code snippets, in-secured third-party libraries and not adopting stable software development practices [7].

Strong authentication mechanism is required in certain IoT systems to prevent attacks namely eavesdropping, impersonation, and mid-attack man-in. However, these type of attacks pose threats to data availability and data confidentiality to the legal user. Therefore, the requirement of IoT systems are secured protocols and strong encryption while there are several scheme of end to end namely Transportation Layer Security (TLS) has an ability to assured the data confidentiality during data transfer has become a key challenge in IoT system is to utilize encryption of light weight because of restricted IoT end device resources. The lack of authentication and insufficient encryption are few significant security issues in IoT during data transfer and privacy interest. Based on the dynamic key generation, the application with smart good transportation has allocated to the customer as per their needs whereas the 4- digit OTP is send from server software to the customer for user authentication that illustrated an instance of dynamic key generation. Anomaly detection has fascinated an important interest over contemporary learning literature because of its applications with broad range of engineering issues [8-11]. However, the unsupervised framework with variable length of anomaly detection issues in which the exploration is to identify the function for deciding whether an every unlabeled sequence of variable length provided in a data set is anomalous or not. Similarly, the attempt of previous malware detection have implemented static analysis and static behavior are the major techniques utilized over code analysis for acquiring data concerned with software behavior [12]. When analyzing these techniques, it unable for detecting files in maintaining the techniques like reverse decompression or packing [13]. This analysis may occur during the software is really running, capturing its behavior is said to be dynamic behavior analysis. Moreover, this technique has the ability to address the issues efficiently which can't be resolved by static detection. In order to detect malicious behavior, there are several experts in the current field has utilized dynamic behavior technique [14, 15]. In general, the unknown malware is classified into known malware using various classification techniques [16].

2. Literature Review

Most research is discussed on the use of ECC for secure authentication. An ECC is described in[17] have been using the sep160r1 curve which are point multiplication for 1.3 MCyclesto, the authors using 8-bit Atmel ATmega for implementing cycle-accurate clones , 16-bit based Texas Instruments such as MSP430, and 32-bit ARM with Cortex-M0 + embedded processors for ECC software implementation [17]. The authors proposed a hardware support including the function of five NIST ECC prime field-based curves in [18]. In [19], the author proposed in a resource-constrained environment by integrated hardware and software design strategies for quick ECC calculations. In [20–22], several other ECC algorithms were introduced through the work of researchers. This is demonstrated by the ECC-based ElGamel encryption scheme. Finite Fields and its implementations in cryptography are provided with effective Program Implementation. The key exchange protocols and cryptographic algorithms are used by thee clients agree to communicate with servers at the beginning of TLS / SSL sessions. RSA, ECDH are the key exchange protocol can be used for private key operations which are violated by calculating the amount of time needed to be completed [23]. Klima et al. [24] recovered from inverting RSA encryption using pre-master key. The known plain text are belongs to attacking approaches that can use permanent

public and private keys to the server. The key exchange has to build sessions, the more vulnerability of TLS / SSL reveals on long-term public and private key from cryptanalysis attacks. At the start of the initial generation of dynamic key, the dynamic key cryptography only executes the key exchange and distributing at once. There's no more key exchange in dynamic key generation. In comparison to TLS / SSL, there is no key sharing in any session of the dynamic key cryptography. However the key exchange risk of dynamic keys is reduced to a minimal.

Despite significant progress made in many machine learning problems through deep learning techniques, there may be a significant shortage of deep learning solutions for detecting anomalies. The researcher presents a detailed overview of the fraud detection approaches focused on deep learning [25]. A wide ranging of the cyber-intrusion detection techniques for deep anomaly detection (DAD) is presented [26]. A description of the 3 IoT and big-data DAD strategies for the Internet was introduced [27]. The methods of deep learning for the identification of video phenomena along with different categories were presented [28]. There are two types of approaches in the detection of malicious behavior dependent on actions namely, detection of static analysis and detection of dynamic analysis [29]. The analysis of static involves removing malicious code using software such as IDA Pro or W32Dasm to disassemble the code. This method does not involve program execution; rather, knowledge about malicious activity is retrieved only by application review. For instance, Wang [30] proposed with related data for comparing code in a malicious behavior file. Next, this method helps to determine the block of code after which, it compares the block of data. However, this approach refers only to code of malicious behavior, which has not recently undergone improvements. Researchers suggested a framework for detecting malicious Android apps by evaluating device activity in a static way [31]. This framework removes static features for detecting malicious activity from vulnerable applications; however, this technique cannot secure computers from intermittent attacks or modified malware. To distinguish malware groups, Vida Ghanaei
[32] proposed a basic block frequencies based on static block evaluation of the malware samples. Hence the actual operation of malware are not considered and only used by static analysis. The samples of the malware get gradually increased at an essential measure and finding out is considered to be complex in earlier detection of Big Data but the significant factor to recognize in collecting actual data is not very accurate. However, the analysis considered in data as an informative which provide a major role for security analyst [33]. Once the proposed framework may resolve the issues and responsibilities which applicable for discovering zero-day malware as earlier that may be assist with potential in developing the data over real-time to find out the zero-day attack of malware and provide participant with earlier corrective measures. There is a comparison among the results of these two classifiers, and it is noted that SVM has the superior ability to detect malware. The findings indicate that SVM achieved the highest 93.03% accuracy to detect malware and benign forms employing 10-fold cross validation [37-42].

3. Proposed system

The Proposed ECC algorithm contain mechanism of double security for securing connected IoT devices over cloud server and progressed for initiating secured TLS handshake or login ID. According to this process, the cloud server is considered to be trusted using provider server’s certificate which gets trusted by devices and once the server name gets matched with applied server’s certificate. After the process of initial handshake or login ID is over, the firmware is downloaded and the signature of firmware gets verified based on the IoT devices. The mechanism of double security is needed for suitable secure in completing the upgrade process is shown in Figure 1 as an architecture for combination of Edge node security with dynamic key generation and malware detection classification. However, the download initiated securely after the completion of confirmation from the devices and thus prevented man in the middle attacks. Once the process of edge node security is progressed through server certification, digital signature has been utilized as an asymmetric cryptography in which the key generation algorithm is using two different set of keys that represented in term of public key and private key. In addition, the private key is identified only for the firmware with computer signing but the public key gets stored over initial

and consequent firmware or within the secure boot of IoT devices. Hence, ECC is used as an asymmetric cryptography algorithm to resource constrained devices too because of it small sized key to the similar level of security.

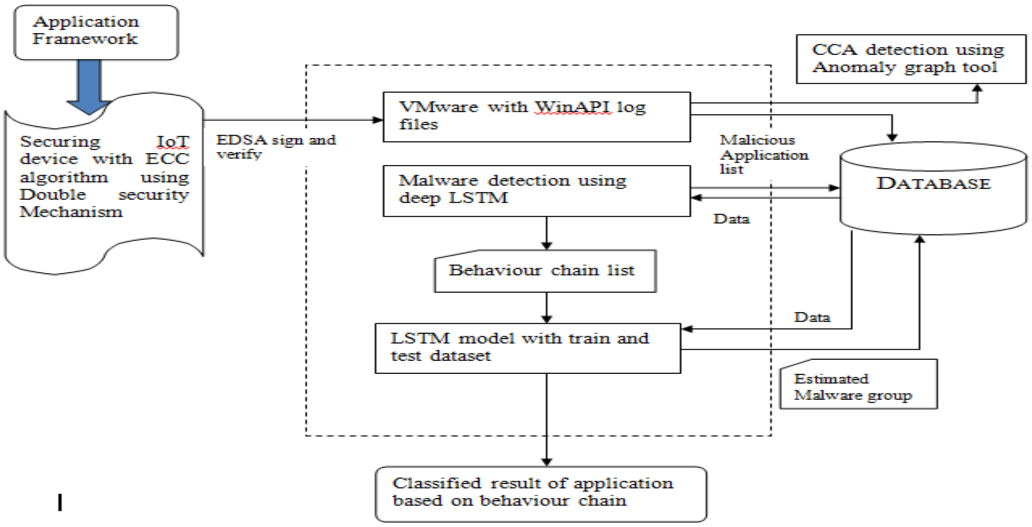

Application Framework
Securing IoT device with ECC algorithm using Double security Mechanism
EDSA sign and verify
VMware with WinAPI log files
Malware detection using deep LSTM
Behaviour chain list
LSTM model with train and test dataset
CCA detection using Anomaly graph tool
Malicious Application list
DATABASE
Data
Data
Estimated Malware group
Classified result of application based on behaviour chain
I

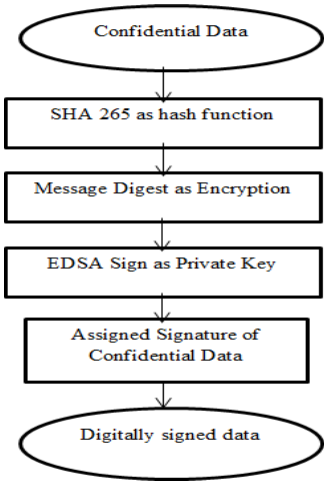
Fig 1 Block diagram for ECC-LSTM architecture

3.1 ECC with double security mechanism in IoT device

The firmware doesn't have ability to sign directly by private key because the firmware size is considerably too large while using that as asymmetric cryptography. According to this factor, the firmware is not made to be signed directly by asymmetric cryptography but login ID or finger print of firmware can be utilized to sign in whereas the fingerprint or login ID can be measured using a SHA256 hashing algorithm. The private computer with the private key calculates a hash on the firmware and signs the hash is shown in the figure 2. In order to get into the server the user need to give his input as mentioned in the IoT devices which can be progressed through hash function like SHA265 for converting the login ID or finger print as an encrypted message digest and a private key gets generated for the message digest before assigning to EDSA sign. The assigned signature of confidential data gets signed by digitally signed data and after the device gets powered, the device software integrity and authenticity need to be verified by cryptographically exchange of digital signatures that has been generated.

Figure 2 Flow diagram for EDSA signing

The hash signature is attached to the firmware and the signed firmware is then by upgrading the firmware has been uploaded to cloud server. During the firmware downloaded by devices from the server of secure cloud and a hash get computed as the data trickles in the firmware [33-37]. The computed hash



signature has been verified by devices with received signature attached to the firmware whereas the firmware get trusted while the two signature gets matched against trust foundation and the validation process is as similar as for the accumulated personal signature to legal binding documents which is authorized and agreed to each transaction. The complete function sets provided from secure boot to authorize the developer for verifying the upgrades of firmware and loadable modules. Moreover, the enhancement of securing the embedded device is done through secure boot by verifying cryptographic in which all recent firmware and software gets authentic and not been obscure hacking or malicious modification.

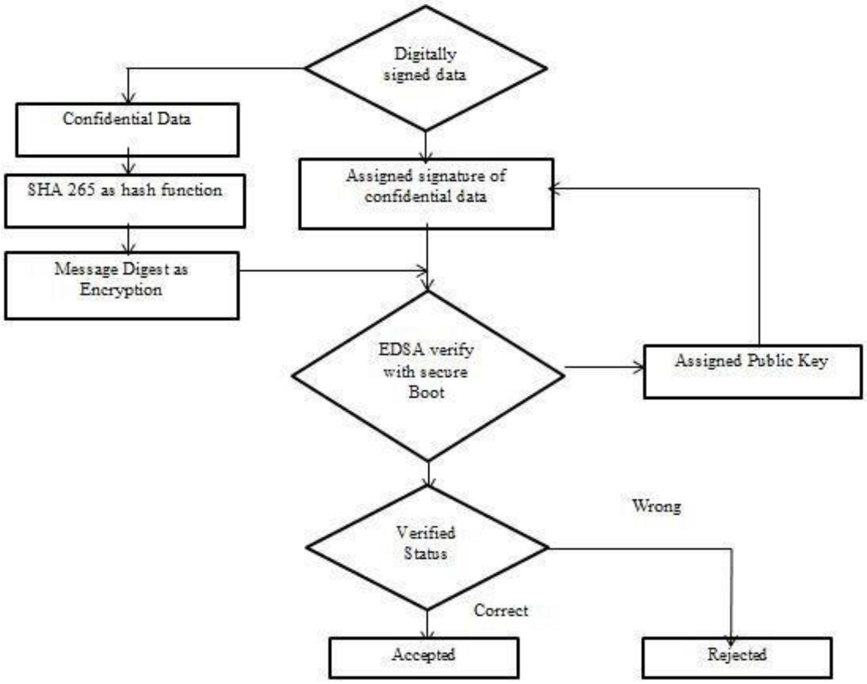


Figure 3 Flow diagram for EDSA verification

3.2 Anomaly and MBD in IoT devised database

There is other kind of attack like anomaly and malware may occur from another hand whereas this proposal has considered those systems continuously collect and monitor performance indicators from its components of database from IoT devices. There are some collected performances indicators have been listed below.

The request of finished HTTP counts and received counts has been extracted from the web server's log files.

The active database connections and active threads counts etc. are extracted from application server by monitoring tools.

The SQL queries number has been accomplished from an application of database management.

Network activities and traffic statistical data in IoTs.

Memory usage, CPU utilization and hard disk transfer rate of IoT devices.

3.2.1 Anomaly Detection using CCA

In this proposed method, the Collective Contextual Anomaly (CCA) technique has been used to detect collective contextual outliers which signify the damaged invariants between multiple sequences of time series. For retrieving the time series contextual information, the sliding window has been utilized to scan via performance indicators. The window size is equivalent to the sequence length which performs as an input for the LSTM network. Hence, this technique has acknowledged for extracting the temporal dynamics present in the time sequences. In order to trigger the process, the invariants have been kept with a value of R-squared which is high while compared to available threshold τm. This has illustrated that accomplished model is required to be verified with various method workloads. Once the uncovered invariants among various time series with undirected graph is utilized for representing relationship between various elements from the global viewpoints which is said to be invariant graphs it is mentioned as G(V,E). For an instance, invariant graph of anomalies representation is shown in figure 4 and every vertex is denoting its time series mode.

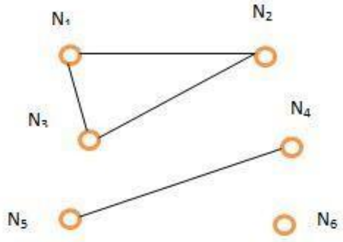


Figure 4 Invariant graph representations

In general, G(V,E) has represented an invariant graph which is shown in figure 4 and in a similar fashion, the representation of anomaly graph gets illustrated as G0(V0,E0) whereas the graph of invariants consists of K vertices, then obviously there may be of K+1 vertices present in the anomaly graph. The top K vertices present in the anomaly graph is as similar as utilized in invariant graph which exists ith node and jth node edge connection represented as $(i,j \leq K)$ and if any invariant to be identified with correspond to invariant graph in which those invariant are predicted to be break soon. Moreover, the ith node has been connected to K+1th node when there is no edge connection of ith node with other node present over invariant graph. Hence, the CCA technique assist to detect the ith data of time series is shown in figure 5 that concoct of 7th node present over an anomaly graph. Thus, all the six nodes represented a univariate time series when the 7th node is virtual node as shown in figure 5 and the predicted two invariants (N1, N2) and (N1, N3) may break soon but a collective anomaly get predicted to occur for time series N6.

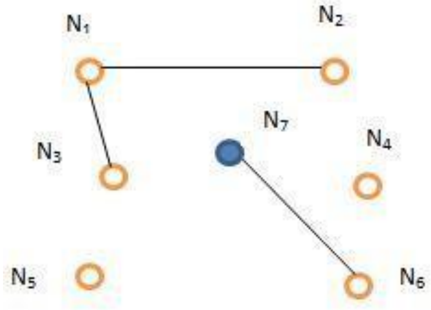


Figure 5 Anomaly graph representations

*3.2.2*MBD using behavior chain

In general, MBD performs as a classification problem with binary and it extracted the behavior analysis form the composed data which is made to be segregated into two cluster namely benign behavior and malicious behavior. However, the collected data get segregated into behavior sequence which is represented as B = {B1, B2, …,Bn}.
Where,
Bi = individual behavior present in the multiple behavior
sets n = Behavior sequence number
Similarly, M= {M1, M2}

Where,
M = Behavior
categories M1 =
Malicious behavior
M2 = Benign behavior
Hence, this proposed technique is to identify an adequate mapping relationship of f(Bi)→Mj, where i ϵ (1,n), j= {1, 2} and f represent classification model mapping function. However, the sample data get collected and trained by suitable LSTM model and conclude at last whether the sample data is malicious or not based on the results of trained models. The process flow of this proposed behavior chain of LSTM train and test data for classification is shown in Figure 6. All sequencers are performed for accomplishing a significant goal whereas every operational step establishes a ―behavior point‖ over process of reaching such goal. According to OS view point, behavior points are considered by a sequence to a particular API functions that may be mentioned as triplet which denoted as X = (R,IO,P),
Where,
R = behavior point call return value
IO = Input and output of the behavior
point P = Purpose of behavior point
However, this attribute has the ability to registry behavior point, file a behavior point, network behavior point etc. The components of IO{IP(P1 : V1; P2 : V2); OP(P3 : V3; P4 : V4)} present over triplet consists of multiple parameters.
Where,
IP = Behavior point input
OP = Behavior point
output

However, the {P1, P2} and {V1, V2} have mentioned the parameters of input and its parameter value for the behavior point correspondingly. Similarly, the {P3, P4} and {V3, V4} have mentioned the parameters of output and its parameter value for the behavior point.

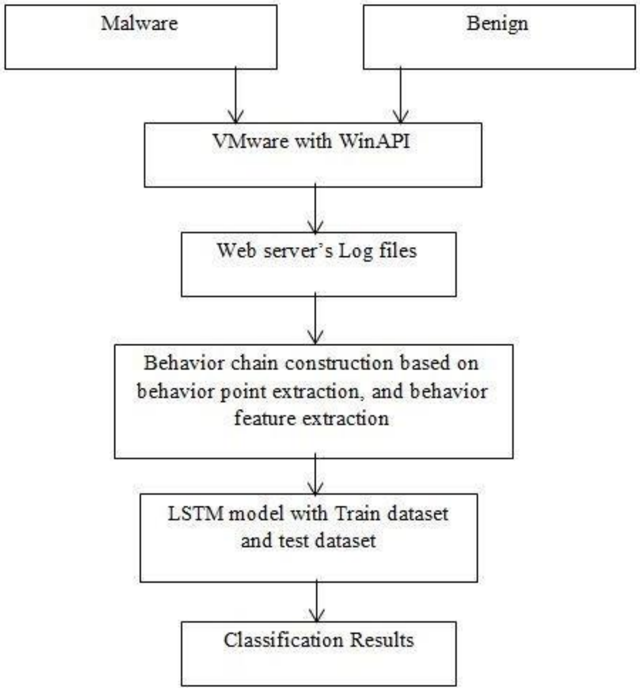

Malware
Benign
VMware with WinAPI
Web server's Log files
Behavior chain construction based on behavior point extraction, and behavior feature extraction
LSTM model with Train dataset and test dataset
Classification Results

Fig 6 Block diagram of MBD and classification

i. Association behavior

A behavioral of association has includes the relationship among behavior points and performances which specifically relates to the timing association. At an instance, the behavior point X1 has mentioned about first call of API while the sequence gets executed, and the behavior point X2 has mentioned about second call of API. However, the parameters and its values present in IP and OP of X2 have been initiated from the previous behavior point X1 which acts as an output of X1 that cast as input for behavior point X2. Hence, the outputs of behavior point X1, X2 are based on the behavior point X1. Thus, the association among the two behavior points is mentioned as:

$X1 \rightarrow X2$

ii. Association behavior Processing

Initially, features of behavior point need to be determined from the data collection is said to be behavior feature extraction whereas the gathered data may involves unwanted data which may establish interference. In order to avoid interference, the data required to be preprocessed and the needed behavior points have been extracted. Subsequently, based on call sequences these behavior points have been clustered into behaviors and the collection of these behavior combinations is made to be behavior chain.

iii. Feature extraction

According to this proposed MBD method, the call format of API has comparatively standardized based on the API names, I/O parameters and its parameter values perform over legitimate monitored log which is expressed as follows:

File{X1(P1 : V1; P2 :V2; P3 : V3,…), X2(P3 : V3; P4 : V4,…),…}

Based on the collective log files, the API extraction is done initially and subsequently the behavior point parameters and values corresponding have been utilized. At the extraction process, it is complicated for extracting an appropriate parameters and its parameter values due to unnecessary interference data occurred from each log file. In order to achieve this goal, text analysis has been involved in usage of string processing techniques. The raw data includes 3 distinct data types namely API names and parameters, API names and API names along with parameters and values. Hence, this proposal majorly focuses on the behavior points and considered only API functions individually but ignored parameters and its parameter values.

The aim of behavior extracted is to integrate the behavior points accomplished by tracking into behaviors like the point of monitored behavior X1, X2 , and X3, into acquired behaviors namely Y = (X1, X2, X3). However, the similar types have been combined while integrating behavior points whereas the behavior point types are based on classification of behavior point. Hence, the extracting behavior points need to be mapped initially with their respective categories of behavior.

iv. Behavior chain process

Moreover, the sequence execution is majorly considered for specific API calls to accomplish a significant goal which act as an associate among the behavior has been represented among the behavior points. Hence, the relationships may be recognized over the relationships of transmission among the behavior points. Every behavior chain contains various behaviors and similarly every behavior consists of several behavior points. When the sequence gets executed for each time, there may be call series to system APIs as far as the sequence attains its desired aim. Thus, the respective behavior chain has been constructed using extraction of API function calls. The behavior chain has been constructed with temporal characteristics for illustrating its disruptive cycle of malicious behavior. A specific behavior is caused if a certain function is known to API call, so the possibility of the subsequent behavior being malicious or benign get evaluated via behavior chain that can be beneficial for planning to detect in advance. According to this study, the collected datasets has been utilized to extracting behavior features and behavior chain. Finally, this behavior chain is implemented in the deep learning familiar model as LSTM for predicting whether the sequence has exhibited malicious behavior or not.

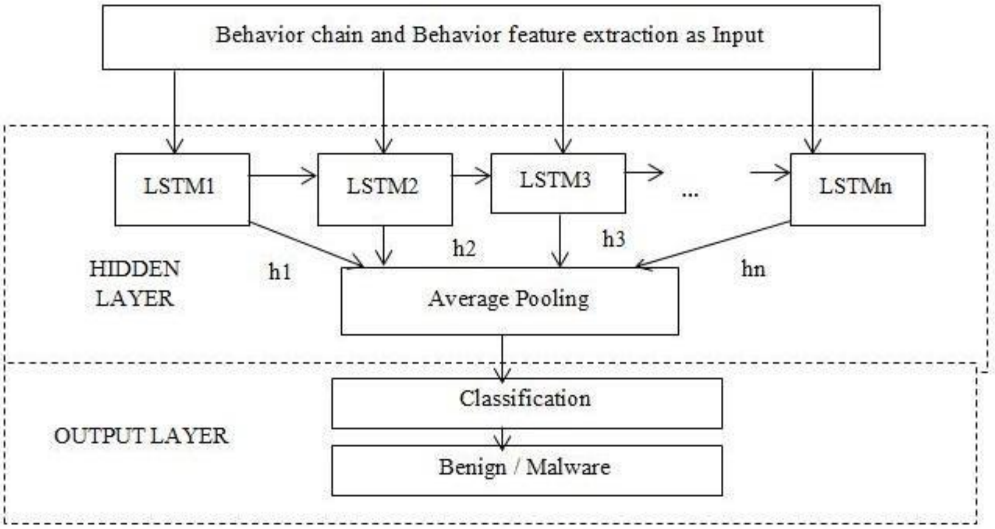


Fig 7 Proposed LSTM model based on Behavior chains

Once the extraction of behavior feature data from log files and building the temporal characteristics of behavior chain, the constructed train model is depending upon these behavior chain and LSTM due to latent anonymous attack caused by anomalous behavior is more suitable while the intruder is frequently try to complicated their behavior of attacks. In general, an individual behavior occurs as common but behavior is associated with its probability for integrating it into abnormal behavior. Hence, this study has analyzed anomalous behaviors from the viewpoint of API calls. Thus, the malicious or normal sequences may both make API calls. The organization of API behavior chain and it utilize the benefits of LSTM model as an efficient recognition technique. The managed behavior chains are considered as an input for LSTM separately to detect and recognize, and the accomplished hidden layer at an individually instant are combined. Subsequently, average pooling is utilized for reducing the dimensions to achieve an expression of transformed data and at last the model creates a classification based on transformed data by classification technique.

4. Results and Discussion

According to this study, the IoT devices used consists of strong device ID and authentication mechanism for installing unique X.509 certificate. The solution of cloud server has been configured for admitting only TLS connections from clients with a X.509 certificated IoT devices as trusted. It is considered to be an authentication of mutual TLS in which the client authenticates the server and similarly server has authenticated the devices of the clients. However, the usage of X.509 certificate to each device needs a certified Authority (CA) manager for signing the certificates. Once the latest firmware's edge node gets updated, the signature assigned to the firmware is required to verify from IoT devices. When firmware is assumed to be trusted until the firmware signature is validated. Hence, the experiment is carried out on a virtual device, the configuration of which is shown in Table 1.

Table 1 configuration of Virtual IoT device

Features	Description
Operating System (OS)	Window 7 professionals
CPU	Intel (R) Xeon(R) CPU E5-2620 V4 @2.10 GHz
Hard disk	60GB
Memory	4GB
VMware	VMware Workstation 12 PRO
Development language and tools	C, Python 2

Nonetheless, this enhanced security testing is necessary because the attacker can able to make an attempt in hacking the server over real time and setting an alternative payload for the firmware as a protected Bus shown in figure 8. Subsequently, the hackers can’t sign the firmware with a verified signature that operates in IoT devices as an enabling factor of security implementation.

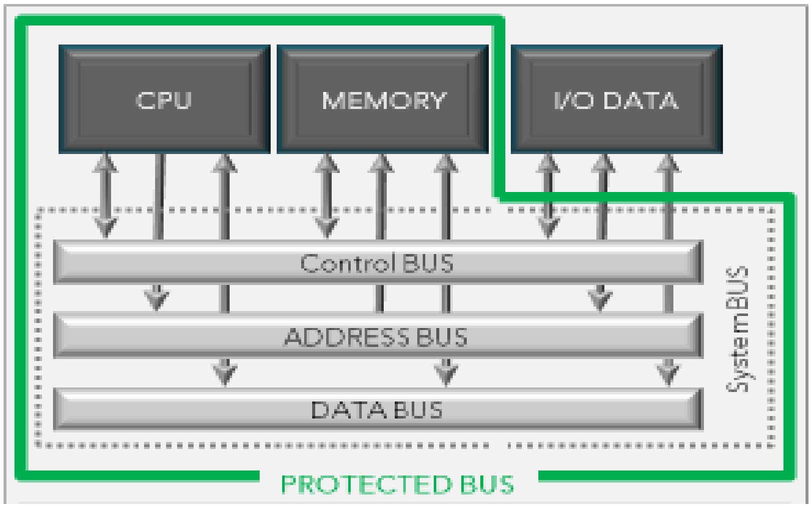
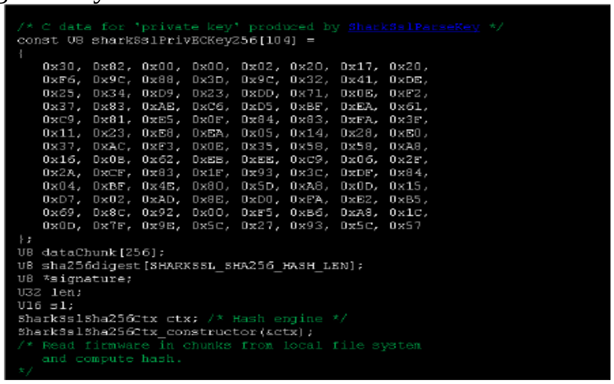


Figure 8 Firmware of Protected Bus

In general, the firmware signing process has been performed over host machine like Windows or Linux when a new release or distribution is being developed. Based on this proposal method, any cryptographic library may be utilized. Hence, the private keys have been dynamically loaded from a file server, or inserted directly into the C program by transforming the key from the format of PEM to SharkSSL by SharkSlParseKey command line tool is shown in figure 9. The authentication method is done through an asymmetric cryptographic library which is compact enough for devices that are limited by memory. The RayCrypto engine involves in SharkSSL as a smart choice because SharkSSL perform as a smallest TLS stack mounted. The dual firmware update includes that the machine store two copies of the firmware upon on device whereas the Secure Boot has been built into the firmware which enables the upgrade logic to use device-residing for TLS and TCP / IP stacks. Hence, the linker creates two firmware versions which can be implemented in lower and upper memory regions simultaneously. The upgrade logic has chosen the right firmware version from the cloud server which to be loaded from online. Thus, the mechanism of Secure Boot can provides an intuitive and efficient way of managing Edge Node firmware updates when maintaining an embedded device's bandwidth and irregular connections without losing security.



```
/* C data for 'private key' produced by SharkSslParseKey */
const U8 sharkSslPrivECKey256[104] =
{
   0x30, 0x82, 0x00, 0x00, 0x02, 0x20, 0x17, 0x20,
   0xF6, 0x9C, 0x88, 0x3D, 0x9C, 0x32, 0x41, 0xDE,
   0x25, 0x34, 0xD9, 0x23, 0xDD, 0x71, 0x0E, 0xF2,
   0x37, 0x83, 0xAE, 0xC6, 0xD5, 0xBF, 0xEA, 0x61,
   0xC9, 0x81, 0xE5, 0x0F, 0x84, 0x83, 0xFA, 0x3F,
   0x11, 0x23, 0xE8, 0xEA, 0x05, 0x14, 0x28, 0xE0,
   0x37, 0xAC, 0xF3, 0x0E, 0x35, 0x58, 0x58, 0xA8,
   0x16, 0x0B, 0x62, 0xEB, 0xEE, 0xC9, 0x06, 0x2F,
   0x2A, 0xCF, 0x83, 0x1F, 0x93, 0x3C, 0xDF, 0x84,
   0x04, 0xBF, 0x4E, 0x80, 0x5D, 0xA8, 0x0D, 0x15,
   0xD7, 0x02, 0xAD, 0x8E, 0xD0, 0xFA, 0xE2, 0xB5,
   0x69, 0x8C, 0x92, 0x00, 0xF5, 0xB6, 0xA8, 0x1C,
   0x0D, 0x7F, 0x9E, 0x5C, 0x27, 0x93, 0x5C, 0x57
};
U8 dataChunk[256];
U8 sha256digest[SHARKSSL_SHA256_HASH_LEN];
U8 *signature;
U32 len;
U16 sl;
SharkSslSha256Ctx ctx; /* Hash engine */
SharkSslSha256Ctx_constructor(&ctx);
/* Read firmware in chunks from local file system
   and compute hash.
*/
```

Fig 9 Transformation of key from PEM format to SharkSSL

In this experiment, the study focused for specific number of malware and benign samples for generating sample datasets. During the benign, the API log files are considered from windows system that contains broadly utilized and familiar software. There are two samples of datasets are considered for this experiments namely Trojans, benign and constructor are the major source to get these malicious Windows system programs have been acquired from https://virusshare.com/ whereas the data is illustrated in table 2.

Table 2 Various malware samples details

Type of Malware	File counts	Sample ratio
Constructor	200	13.3
Benign	580	37.5
Trojan	100	6.5

According to this study, the samples collected consists of 300 malware and 580 benign has been utilized as the dataset has shown in Table 2 with several type of malware sample and the file count and sample percentage in every category as raw data. However, the dataset provided from the site is comparatively small but the dataset can be expanded in accordance with every malicious type present over malware dataset for assuring all kind of malware features remains flawless. Hence, data sample get segregated randomly with 70% of data for every type from malicious sample set and similarly 30% from the benign sample dataset have been acquired.

Moreover, the proposed ECC-LSTM is compared with the traditional deep learning model and proved its behavior chain efficiency and the classification results accuracy and other metrics has illustrated and advantage and efficiency of behavior chain with the summarized evaluated metric like False Positive

Rate (FPR), accuracy and False Negative Rate (FNR) along with other traditional models namely Conventional Neural Network (CNN), Support Vector Machine (SVM) is shown in Table 3.

Table 3 comparison of evaluated metric with proposed LSTM and existing model

Experimental model	FPR in%	Accuracy in %	FNR in %
CNN	3.85	96.15	5.97
LSTM	1.04	98.66	3.41
SVM	3.13	93.9	14.51

In this experiment, the detection results of malicious behavior have been compared with traditional processing models and the succeeded observations is obtained in the Table 3 and Figure 10. In the case of FPR, the percentage value need to be low for better FPR whereas LSTM is 1.04% is comparatively better than other two CNN and SVM with 3.85% and 3.13% and in FNR, the proposed LSTM is comparatively lower than SVM and CNN. The accuracy value of SVM is 93.9% and 96.15% in CNN which are comparatively lower than proposed LSTM model that has 98.66%.

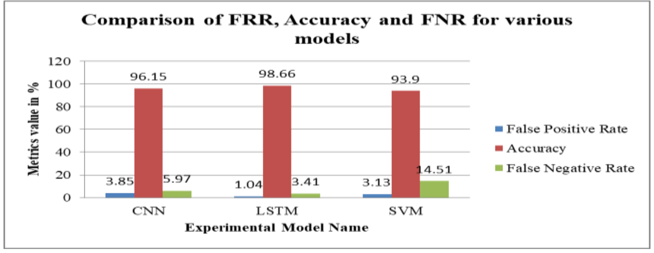


Figure 10 Comparison of FRR, Accuracy and FNR for various models

5. Conclusion

In this paper, the proposed method is hybrid methodology considering edge node computing with ECC algorithm with double security mechanism for the firmware of IoT devices. The secure boot plays a major role in upgrading the firmware using double firmware upgrade that make the firmware upgrade process while the IoT devices is in execution which represent upgrade can be done during real time. The digital signature used in asymmetric cryptography for the resource constraint device is ease due to smaller key size. Similarly the LSTM model which one of the familiar deep learning model has been utilized for real time application in detecting anomaly and MBD. In order to detect the CCA over time series data based on LSTM model and it get progressed by developing a tool named anomaly for representing the detection of CCA. In the case of MBD, API call sequences with behavior feature extraction and behavior chains with temporal characteristics have been constructed which is made to be trained the LSTM network based on the behavior point. However, the behavior points needed to the experiment get extract from an application monitored log files using monitoring the sequence of API call that finally utilized to provide benign or malware sequence classifications. Hence, the results have illustrated that the model's accuracy of the proposed model ECC-LSTM is 98.66% while compared to the existing model like CNN and SVM. Thus, this study has analyzed an individual APIs but not endeavor for considering the influence of the parameters or its parameter values which act as an input or output by those APIs on MBD for IoT devices.

References

[1] D. Linthicum, ``Responsive data architecture for the Internet of Things," Computer, vol. 49, no. 10, pp. 72_75, 2016.

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, ``A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," IEEE Internet Things J., vol. 4, no. 5, pp. 1125_1142, Oct. 2017.

[3] J. A. Stankovic, ``Research directions for the Internet of Things," IEEE Internet Things J., vol. 1, no. 1, pp. 3_9, Feb. 2014.

[4] J. Wu and W. Zhao, ``Design and realization of WInternet: From net of things to Internet of Things," ACM Trans. Cyber-Phys. Syst., vol. 1, no. 1, pp. 2:1_2:12, Nov. 2016. [Online]. Available: http://doi. acm.org/10.1145/2872332.

[5] H. Ning, H. Liu, L. T. Yang, Cyberentity security in the internet of things, Computer 46 (4) (2013) 46{53. doi:10.1109/MC. 2013.74.

[6] S. Sicari, A. Rizzardi, L. Grieco, A. Coen-Porisini, Security, privacy and trust in internet of things: The road ahead, Computer Networks 76 (2015) 146- 164.

[7] M. Patton, E. Gross, R. Chinn, S. Forbis, L. Walker, H. Chen, Uninvited connections: A study of vulnerable devices on the internet of things (iot), in: 2014 IEEE Joint Intelligence and Security Informatics Conference, 2014, pp. 232{235. doi:10.1109/JISIC.2014.43.

[8] V. Chandola, A. Banerjee, and V. Kumar, ―Anomaly detection: A survey,‖ ACM Comput. Surv., vol. 41, no. 3, p. 15, Jul. 2009. doi: 10.1145/1541880.1541882.

[9] N. Görnitz, L. A. Lima, K. Müller, M. Kloft, and S. Nakajima, ―Support vector data descriptions and k-means clustering: One class?‖ IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 9, pp. 3994–4006, Sep. 2018.

[10] Z. Ghafoori, S. M. Erfani, S. Rajasegarar, J. C. Bezdek, S. Karunasekera, and C. Leckie, ―Efficient unsupervised parameter estimation for one-class support vector machines,‖ IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 10, pp. 5057–5070, Oct. 2018.

[11] Q. Chen, R. Luley, Q. Wu, M. Bishop, R. W. Linderman, and Q. Qiu, ―AnRAD: A neuromorphic anomaly detection framework for massive concurrent data streams,‖ IEEE Trans. Neural Netw. Learn Syst., vol. 29, no. 5, pp. 1622–1636, May 2018.

[12] Han, K.S., Kim, I.K., Im, E.G.: Malware classification methods using API sequence characteristics. Lecture Notes in Electrical Engineering(LNEE). 120, 613–626 (2012).

[13] Huang, J., Swindlehurst, A.L: Secure communications via cooperative jamming in two-hop relay systems. In: IEEE Globecom, pp. 1–5 (2010).

[14] Berlin, K., Slater, D., Saxe, J.: Malicious behavior detection using windows audit logs. In: Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security(AISec), pp. 35–44. ACM (2015).

[15] Mosli, R., Li, R., Yuan, B., Pan, Y.: Automated malware detection using artifacts in forensic memory images. In: Technologies for Homeland Security (HST), pp. 1–6. IEEE (2016).

[16] Rieck, K., Laskov, P.: Linear-time computation of similarity measures for sequential data. J. Mach. Learn. Res. 9(9), 23–48 (2008).

[17] Wenger, E., Unterluggauer, T., Werner, M.: 8/16/32 shades of elliptic curve cryptography on embedded processors. In: Progress in Cryptology INDOCRYPT 2013. Lecture Notes in Computer Science. Springer International Publishing, vol. 8250, pp. 244–261 (2013)

[18] Alrimeih, H., Rakhmatov, D.: Fast and flexible hardware support for ECC over multiple standard prime fields. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 99, 1–14 (2014)

[19] Höller, A., Druml, N., Kreiner, C., Steger, C., Felicijan, T.: Hardware/software co-design of elliptic-curve cryptography for resource-constrained applications. In: 51th ACM/EDAC/ IEEE Design Automation Conference (DAC), June 2014.

[20] Aigner, H., Bock, H., Hütter, M., Wolkerstorfer, J.: A low-cost ECC coprocessor for smartcards. In: Cryptographic Hardware and Embedded Systems (CHES). Lecture Notes in Computer Science, vol. 3156, pp. 107–118. Springer, Heidelberg (2004).

[21] Hein, D., Wolkerstorfer, J., Felber, N.: ECC is ready for RFID a proof in silicon. In: Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 5381, pp. 401–413. Springer, Heidelberg (2009).

[22] Plos, T., Hutter, M., Feldhofer, M., Stiglic, M., Cavaliere, F.: Security enabled near-field communication tag with flexible architecture supporting asymmetric cryptography. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 21(11), 1965–1974 (2013).

[23] P. C. Kocher, Timing attacks on implementations of Di±e-Hellman, RSA, DSS and other systems," Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, LNCS 1109, pp. 104-113, Springler Verlag, 1996.

[24] P.Sakthi Shunmuga Sundaram et al. ―Smart Clothes with Bio-sensors for ECG Monitoring‖, International Journal of Innovative Technology and Exploring Engineering, Volume 8, Issue 4, (2019), pp. 298-30.

[25] Aderemi O Adewumi and Andronicus A Akinyelu. A survey of machine-learning and nature-inspired based credit card fraud detection techniques. International Journal of System Assurance Engineering and Management, 8(2): 937–953, 2017.

[26] Donghwoon Kwon, Hyunjoo Kim, Jinoh Kim, Sang C Suh, Ikkyun Kim, and Kuinam J Kim. A survey of deep learning-based network anomaly detection. Cluster Computing, pages 1–13, 2017.

[27] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani. Deep learning for iot big data and streaming analytics: A survey. arXiv preprint arXiv:1712.04301, 2017.

[28] B Ravi Kiran, Dilip Mathew Thomas, and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. arXiv preprint arXiv:1801.03149, 2018.

[29] Uppal, D., Sinha, R., Mehra, V., Jain V.: Malware detection and classification bases on extraction of API sequences. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics(ICACCI), pp. 2337–2342. IEEE (2014)

[30] Wang, Z., Pierce, K., McFarling, S.: BMAT—a binary matching tool for stale profile propagation. The Journal of Instruction-Level Parallelism(JILP). 10(2), 23–25 (2000)

[31] Fereidooni, H., Conti, M., Yao, D., Sperduti, A.: ANASTASIA: android malware detection using static analysis of applications. In: Proceedings of the 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–5 (2016).

[32] Hansen, S.S., Larsen, T.M.T., Stevanovic, M., Pedersen, J.M.: An approach for detection and family classification of malware based on behavioral analysis. In: Proceedings of the International Conference on Computing, Networking and Communications (ICNC), pp.1–5 (2016).

[33] V. R. Niveditha, T. V. Ananthan, S. Amudha, D. Sam, and S. Srinidhi, ―Detect and classify zero day Malware efficiently in big data platform,‖ Int. J. Adv. Sci. Technol., vol. 29, no. 4 Special Issue, pp. 1947–1954, 2020.

[34] Natrayan, L., and M. Senthil Kumar. Optimization of squeeze casting process parameters on AA2024/Al2O3/SiC/Gr hybrid composite using taguchi and Jaya algorithm, International Journal of Control and Automation, Vol.13, No.2s, (2020), pp.95-104.

[35] K. Amandeep Singh and T. V. Ananthan, Research Challenges on Big Internet of Things Data Analytics, Journal of Computational and Theoretical Nano science, Vol. 16, (2019), 2113–2116,

[36] V. R. Niveditha and Ananthan TV, ―Improving Acknowledgement in Android Application‖, Journal of Computational and Theoretical Nano science. 16, (2019), pp. 2104–2107

[37] L. Natrayan et al., Effect of graphene reinforcement on mechanical and microstructure behavior of AA8030/graphene composites fabricated by stir casting technique, AIP Conference Proceedings, 2166, (2019), pp. 020012.

[38] S. Velliangiri, P. Karthikeyan & V. Vinoth Kumar (2020) Detection of distributed denial of

service attack in cloud computing using the optimization-based deep networks, Journal of Experimental & Theoretical Artificial Intelligence, DOI: 10.1080/0952813X.2020.1744196

[39] Vinoth Kumar V, Karthikeyan T, Praveen Sundar P V, Magesh G, Balajee J.M. (2020). A Quantum Approach in LiFi Security using Quantum Key Distribution. International Journal of Advanced Science and Technology, 29(6s), 2345-2354.

[40] Umamaheswaran, S., Lakshmanan, R., Vinothkumar, V. et al. New and robust composite micro structure descriptor (CMSD) for CBIR. International Journal of Speech Technology (2019), doi:10.1007/s10772-019-09663-0

[41] Karthikeyan, T., Sekaran, K., Ranjith, D., Vinoth kumar, V., Balajee, J.M. (2019) ―Personalized Content Extraction and Text Classification Using Effective Web Scraping Techniques‖,
International Journal of Web Portals (IJWP), 11(2), pp.41-52

[42] Maithili, K , Vinothkumar, V, Latha, P (2018). ―Analyzing the security mechanisms to prevent unauthorized access in cloud and network security‖ Journal of Computational and Theoretical Nanoscience, Vol.15, pp.2059-2063.