Trust Management for Deep Autoencoder based Anomaly Detection in Social IoT

Rashmi M R[1], C Vidya Raj[2]
Research Scholar, VTU, Belagavi[1]
Department of Computer Science and Engineering, NIE, Mysore, India[1,2]

***Abstract*—Social IoT has gained huge traction with the advent of 5G and beyond communication. In this connected world of devices, the trust management is crucial for protecting the data. There are many attacks, while DDOS is the most prevalent BotNet attack. The infected devices earnestly require anomaly detection to learn and curb the malwares soon. This paper considers 9 IoT devices deployed in a Social IoT environment.We introduce a couple of attacks like Bash lite and Mirai by compromising a network node. We then look for traces of malicious behavior using AI algorithms. The investigation starts from a simple network approach - Multi-Layer Perceptron (MLP) then proceeds to ML - Random Forest (RF). While MLP detected the malicious node with an accuracy of 89.39%, RF proved 90.0% accurate. Motivated by the results, the Deep learning approach - Deep autoencoder was employed and found to be more accurate than MLP and RF. The results are encouraging and verified for scalability, efficiency, and reliability.**

Keywords*—*Social IoT; trust management; anomaly detection; DDoS; deep autoencoder

I. Introduction

IoT is a disruptive network technology that has advanced quickly over the past ten years in every technology field, including smart cities, satellites, smart homes, smart businesses, smart transportation, and smart healthcare [1]-[4]. It consists of several IoT devices (Things) that may gather and share data through the conventional internet thanks to their various sensors, actuators, storage, computing, and communication capabilities [5]. The industry's security concerns resulting from the enormous range of IoT devices and vendors. On the off chance that security and protection are not accommodated their organizations and information, partners are probably not going to broadly embrace IoT innovations. Recent cyber security reports [6] have revealed that assaults against IoT settings have increased in frequency due to the IoT ecosystem's expanding attack surface, which extends from the edge to the cloud [7]. Therefore, a significant ongoing problem for engineers in this industry is designing and creating secure IoT systems [8]. The sensitive nature of the data collected and processed within the IoT network necessitates security from potential breaches. As the first line of defense against potential security attacks [9]-[10] on weak devices [11], like distributed denial of service (DDoS) attacks [12], various security mechanisms are currently used to protect sensitive data. These mechanisms include firewalls, authentication protocols, encryption methods, antivirus software, and more. Such assaults are carried out against another network entity, such as a business or a government, by a collection of infected machines (bots) that are part of a botnet and are under the attacker's control via a C&C (Command and Control) server. Due to the extensive use of data, several new anomalies—original and mutations of previously observed anomalies—are often produced.

The Compromise of IoT devices and their enrollment into IoT botnets under attackers' control is one of the main threats to IoT networks and devices. Well-known IoT botnets like BASHLITE and Mirai continue to pose substantial DDoS risks, according to the conclusions of a Hundred of active command and control (C & C) servers are included in the Distributed Denial of Service (DDoS) report for the first quarter of 2021 [13]. Due to the IoT's primary characteristics, which must be considered: heterogeneity, scalability, and limited resources, mitigating such threats could be very difficult (power, memory, and processor). As a result, creating solutions for IoT environments that can detect aberrant behaviors and assaults has emerged as a major problem in the field of IoT cyber security and a hot topic for researchers. Subsequently, an IoT organization can profit from extra protection from security attacks thanks to a anomaly identification framework that can act as a second line of guard. Furthermore, enterprises in this market sector are primarily focused on fusing IoT technology with other slashing technologies like AI (AI algorithms are used for data processing and analysis), Big Data (handling a huge amount of information from IoT devices), or 5G connectivity (mobility and broadband links for IoT sensors). Since it merges AI with IoT, the AI of Things (A-IoT), a disruptive technology that aims to analyze data to make autonomous and automated decisions on IoT networks, is receiving special attention [14].

To increase anomaly detection accuracy, researchers have recently looked into machine learning (ML) and deep learning (DL) techniques. Studies have shown that both ML and DL approaches are useful for extracting characteristics from network traffic that can be used to classify the traffic as benign or abnormal [15]. The DL has shown effective at learning relevant attributes from the raw data because of its deep model, which provides a variety of abstractions for learning intricate features for precise predictions [16]. Due to the huge volume of data generated by IoT devices, these characteristics of DNN have made it a suitable methodology to be adopted for anomaly detection schemes created for IoT networks [17]. Larger networks, however, might find these solutions impractical. In addition, because a separate model needs to be updated and maintained for individual IoT sensor, using several auto encoders might make network security challenging to establish and administer. We offer three methods. Deep auto-encoder, Random Forest, and Multilevel Perceptron for identifying anomalies in IoT networks by observing and analysing the innocuous "snapshots" of behaviour from each IoT device, we suggest a deep auto-encoder-based anomaly

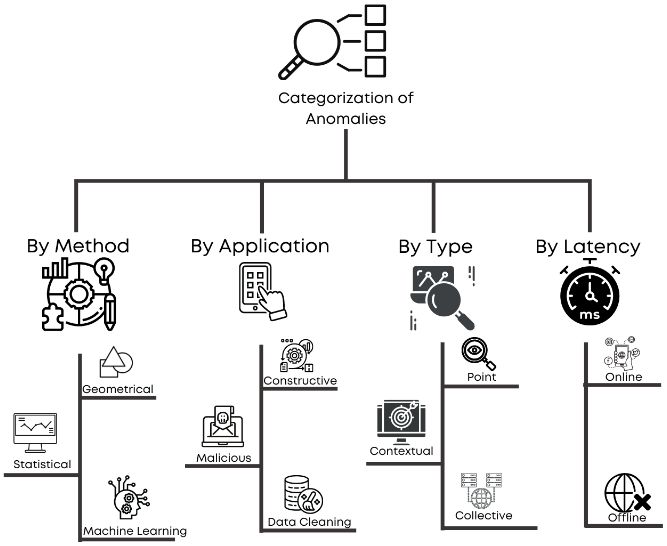

Fig. 1. Categories of anomaly in IoT

detection system as a viable method for identifying botnet attacks. The experiment is built on a test bed network of nine IoT devices and simulates the BASHLITE and Mirai botnets, two well-known botnets. Auto-encoders well defined for each IoT gadget is used to gain proficiency with the regular traffic properties and to caution when they can't recreate the harmless traffic samples.

The main contribution of the work is three-fold:

• First, a brief review of the constraints and vulnerabilities of Social IoT networks, various attacks, and anomalies is presented.

• Next, we deploy 9 IoT devices in a Social set-up while injecting one node with BASHLITE and Mirai infection resulting in the DDOS attack.

• Finally, we employ ML, NN, and DL-based approaches - RF, MLP, and Deep Auto-encoder- to detect anomalies accurately and ascertain the results for scalability, efficiency, and reliability.

The following is how the paper is structured. Section II introduces anomaly detection and discusses numerous sorts of anomalies in the IoT context. Furthermore, it situates the topic within the framework of DDoS attacks and gives a taxonomy of DDoS attacks. Section III discusses anomaly detection strategies and forms of anomaly attacks in the social IoT and finishes with the deep autoencoder as a solution for anomaly identification. The proposed framework for anomaly detection and the evaluation metric to validate the performance of the proposed method are then presented (Section IV). We then examine the intriguing results that demonstrate the superiority of the suggested strategy over rival schemes in Section V. The final remarks are discussed in Section VI.

II. Background

A. Categorization of IOT Anomalies

An anomaly is a data point in a modelled system that is not aligned to the normal behaviour. Rare occurrences or observations known as anomalies differ dramatically from typical behaviour or patterns seen in a single data point or throughout the full dataset. A aim of algorithm will probably find an irregularity's event and characterize/gather its objective in light of the fact that in principle, anomalies are brought about by outside powers like sensor breakdown or outer assault. The estimation structure that best matches the expected information conduct is fundamental in the twofold order of a peculiarity. Additionally, each application needs a unique detection approach due to the complexity of numerous circumstances [19].

Taking into account the arrangements from earlier examinations like Fahim and Sillitti [20] and Cook et al. [21], an IoT peculiarity discovery approach is partitioned into four classes. According to the problem they address, how they are used, the kind of method used, and the algorithm's latency, they are divided into different categories. Fig. 1 presents an example overview of the four groups.

(i) By Method: The methods can use machine learning, statistics, or geometrical methods. Geometrical approaches are based on the presumption that the anticipated and abnormal data are separated when distance- and density-based representations of a given dataset are used. The reasoning behind detachment or density-based approaches is that peculiarities arise in scanty districts in a bunch of data of interest. These procedures classify irregularities utilizing a static or dynamic magnitude value (t) on the assessed distance (d), which is given as:

$$d = \left\{ \begin{array}{c} < t, Normal(underthreshold) \\ > t, Anomaly(abovethethreshold) \end{array} \right\} \quad (1)$$

(ii) By Application: The three methods that an application uses to categorize anomalies are data cleansing, data destruction, and constructive categorization. The world of elderly people's everyday activities to ensure safe and its evaluation of the efficiency of multilayer perceptron (MLP), SVM classifiers, and k-nearest neighbours (KNN), offer value when applications are constructive or positive in nature.

(iii) By Anomaly Type: The situation-specific type, such as the point, contextual, and collective, is one of the most frequently observed types. If just one piece of data deviates from the norm, it qualifies as an anomaly. The identification of fraud with credit cards is one instance. An incident that might be seen as abnormal in a certain context is called a contextual anomaly. The last kind of anomaly, collective anomaly, examines the complete dataset, unlike a point or contextual anomaly.

(iv) By Latency: Whether a detection technique is conducted immediately during the data collecting stage or after it has been stored depends on its latency and scalability [18]. An online technique can serially analyse data with a single point of information or window without having access to the

TABLE I. ATTACK TYPE CLASSIFICATION FOR IOT

Sl.No	IoT Attack	Description
1	Dos	This type of attack involves the deliberate sending of bugs or packets to render resources unavailable to hosts connected to the internet.
2	Data Type probing	This exploit involves a hostile attacker writing an unintended data type .
3	Malicious Control	This type of attack allows the attacker unapproved access to the user's system
4	Malicious Operation	In general, the malware was a factor in this attack. Attacker engages in bogus activity on a system that has been authenticated
5	Scan (SC)	In this assault, equipment sporadically scans the framework for information to obtain, which might ruin information
6	Spying (SP)	In this attack, the attacker targets the system's weak spots and gains access via a backdoor to steal sensitive data
7	Wrong Setup	It is an assault, wherein culprit deliberately sends bugs or parcels for making asset inaccessible for the host associated with the web
8	Normal	This attack may have been planned or unintentional, but it might still cause harm by upsetting the system.

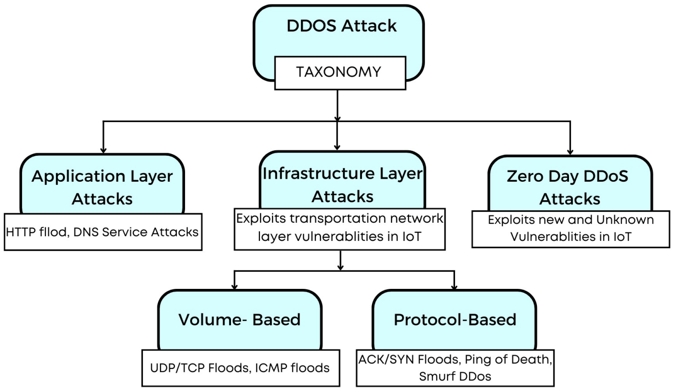
DDOS Attack
TAXONOMY
Application Layer Attacks
HTTP fllod, DNS Service Attacks
Infrastructure Layer Attacks
Exploits transportation network layer vulnerablities in IoT
Zero Day DDoS Attacks
Exploits new and Unknown Vulnerablities in IoT
Volume- Based
UDP/TCP Floods, ICMP floods
Protocol-Based
ACK/SYN Floods, Ping of Death, Smurf DDos

Fig. 2. Taxonomy of DDoS attacks

complete input. Traditional and online geometry and statistical methodologies include the distance-, density-, and angle-based approaches listed above. Offline algorithms, however, have complete access to the information. They employ difficult, computationally expensive, sophisticated methods to solve the problem in a reasonable amount of time. Anomalies are brought on by external forces like sensor malfunction or an external attack. The various types of attacks in IoT have been summarized in Table I. The most common kind of assault that may be launched against any application is a DDoS attack. DDoS attacks are the loudest kind of cyber attacks.

B. Distributed Denial of Service Attacks

The DDoS attack is undertaken to overwhelm the target and interrupt services, as the name suggests. IoT devices are highly suited for the DDoS attack because it needs a lot of devices to conduct an attack. Users won't recognise that the gadget is compromised, as is typically the case. There is a pressing need to identify assaults quickly in order to remove affected devices as the number of IoT devices grows. BASHLITE and Mirai employed IoT devices as Botnets in a large DDoS attack, and other similar attacks have also occurred [22].The various types of DDoS attacks have been illustrated in Fig. 2.

DDoS attacks have demonstrated a variety of attacking strategies over the years, and a variety of potential attacks are continuously being tested. IoT-specific DDoS attack strategies are not much different from conventional DDoS attack strategies. They use similar methods to take advantage of flaws in both IoT devices and conventional systems. However, because of the heterogeneity present in IoT devices, DDoS assaults targeted specifically at IoT are more varied and complex. We will use the fundamental layered architecture of an IoT network to categorise DDoS assaults in this section.

Three different types of DDoS attacks are illustrated in a comparative analysis in Table II. DDoS assaults have affected well-known service companies like Amazon Web Services. AWS, Cloud are, KrebsOnSecurity, and other security service providers against similar assaults are also DDoS attack victims. Therefore, assaults on these significant institutions affect enterprises financially and reputationally. In a DoS attack, the attacker makes bogus requests using the target's resources in an effort to disrupt the target's services. DDoS involves simultaneous demands coming from several sources. DDoS attack mitigation becomes challenging as a result. There are many different types of DDoS attacks, such as Teardrop, Smurf, TCP SYN Flood, Smurf, Teardrop, Botnet attack and Ping of Death. DDoS assaults can also be categorised as amplification and reflection assaults. The request and response sizes are equal in a reflection attack [23], however in an amplification assault, the response size is significantly larger than the request size [24].

1) Compromising an IoT device (BoT) and BoTNets: Due to the inherent characteristics of botnets, namely the existence of widely dispersed peers and C&C servers across the Internet with masked communication techniques, there is no secure strategy that can be utilized to shut down all bot movement without disrupting real traffic. Bot malware like Mirai actively searches the network for weak points, hunting for devices that allow unauthenticated access or that use weak or default credentials. After the defence is broken, a concise bootstrap script is executed, which downloads the whole program from the C&C. Other methods of spreading the dangerous bot code include the widespread use of phishing emails and freeware promotions to trick people into downloading it on their PCs.Making sure the bot binaries avoid antivirus programmes, which often employ signature-based detection techniques, is just as crucial as the bot binaries' distribution mechanism.It was found that Storm was doing this by repeating the encoding its un authorized two times every hour. Although, as IoT sensors lack the processing capacity required to run sophisticated anti-virus software, botnets that target them may conveniently ignore this complexity.

The rallying phase, which occurs after infection, entails

TABLE II. ATTACK TYPE CLASSIFICATION FOR IOT

Variety of Attack	Attacker's Objective	Size Measured in	Examples	Existing Countermeasure to the Attack
Volumetric Attack	Take up the entire bandwidth between the target and the internet	Bits per Second (bps)	UDP, TCP, DNS floods and amplification of NTP	On-demand scalable scrubbing centres get the rerouted traffic so they can handle it.
Protocol Based Attack	To use server, firewall, and load balancer resources	Packets per Second (pps)	Flooding the TCP SYN, Death Ping, Smurf Attack	Recognizable proof method is utilized regularly to separate among authentic and ill-conceived traffic to impede the assault prior to arriving at the objective server.
Application Layer Attack	To exhaust target resources	Requests per Second (rps)	HTTP Flood, DNS Flood	These are by and large sluggish assaults and relieved by recognizing bot conduct utilizing manual human tests and comparative procedures

developing a hidden method for receiving instructions from the CC [25]. This stage's fundamental objectives are to hide the address of the C&C and ensure that any orders shipped off the bots are encoded. Among the components are the "fast flux" strategy (Tempest), which rapidly pivots the C&C server's tends to behind a DNS name, and the utilization of Domain-Generation Algorithm (DGA) [26]-[27], Which require each recently tainted machine to attempt to determine haphazardly created area names to recognize its C&C. Later changes enjoy taken benefit of distributed correspondence, which further clouds the C&C [28]-[29].

III. ANOMALY DETECTION TECHNIQUE

Anomaly or outlier detection problems can be used to frame the task of identifying an assault. This is predicated on the idea that malware and regular network traffic would differ in certain ways, allowing an algorithm to distinguish between the two. In this piece of research, we employ deep learning techniques. The utilization of sophisticated artificial neural networks architecture, that is modeled after the human brain and compute in a completely different way from conventional digital methods, is the cornerstone of deep learning approaches.To learn the weights of the network and create a model that can distinguish between assaults and normal behaviour, deep networks in a NADS technique need to know something about the valid data class.

The utilization of autoencoders in non-linear cooperation and for applications requiring network traffic highlights is an expected system for anomaly identification. Autoencoders can learn more effectively with less training data when depth is used because it lowers the computational cost of modelling functions [30]. These affirmations propelled us to test a Deep autoencoder model for irregularity detection in IoT system. An autoencoder is a neural network-based unsupervised learning model that has been trained to reconstruct the input into the output. It is made up of two parts: an encoder and a decoder. The encoder is used for input, and the decoder is used for output (code).

A. Deep Autoencoder

Deep Auto Encoder (DAE) is a tool for unsupervised learning of effective coding. The simplest DAE architecture consists of an input layer, several hidden layers, and an output layer that contains the same number of neurons for reconstruction as the input layer. It becomes a deep autoencoder when both the encoder and the decoder, the two parts of the autoencoder, are deep networks. The decoder's layers are inverted, but both devices share a similar construction. A deep autoencoder features of Deep autoencoder are:

1) A array X representing n dimension input data, where $X = (X_1, X_2, ...X_n)$

2) Fig. 3 shows several hidden layers that stand in for various encoding and decoding levels. These layers produce an irregular illustration of the data input and reconstruct it for the output layer.

3) The array $X^{'} = (X_1^{'}, X_2^{'}, ...X_n^{'})$ is an output layer. The output, which is a recovered copy of the input data, is the same size as the input.

4) In addition to weights and biases, an activation function. An activation function is used by each neuron in a layer to determine its output based on the weighted sum of its input.

There are two types of activation functions utilised in DL models: direct capabilities and non-straight capabilities. The most famous nonlinear actuation capabilities are the sigmoid (calculated), exaggerated digression, and corrected direct unit (ReLU). The exaggerated digression capability is generally used in two-class characterization, the sigmoid capability is explicitly utilized while determining the result as a likelihood, and the ReLU capability is the most often utilized in basically all profound brain organizations. In our examination, we applied the Sigmoid capability (1) to the last layer of the decoder and the ReLU capability (2) to each secret layer of the autoencoder.

$$S(x) = \frac{1}{1 + e^x} \tag{2}$$

$$R(x) = \begin{Bmatrix} x, x > 0 \\ 0, x \leq 0 \end{Bmatrix} \tag{3}$$

A machine learning model's learnable parameters are weights and biases. The biases and weights are assigned to the inputs before they are passed across neurons. While biases guarantee that neuron activation will still occur even if all the inputs are zeros, loads show the amount of impact the info possesses on the result. In a profound autoencoder, layer l is addressed by W_{ij}^l which addresses the weight applied to the connection between hub j of layer $l-1$ and hub of layer l, and $b_i^{(l)}$,which addresses the predisposition connected with the hub. Following formula is used to determine neuron i's output value from layer l.

$$O_i^l = F(\sum(W_{ik}^{(l)}) + b_i{}^l \tag{4}$$

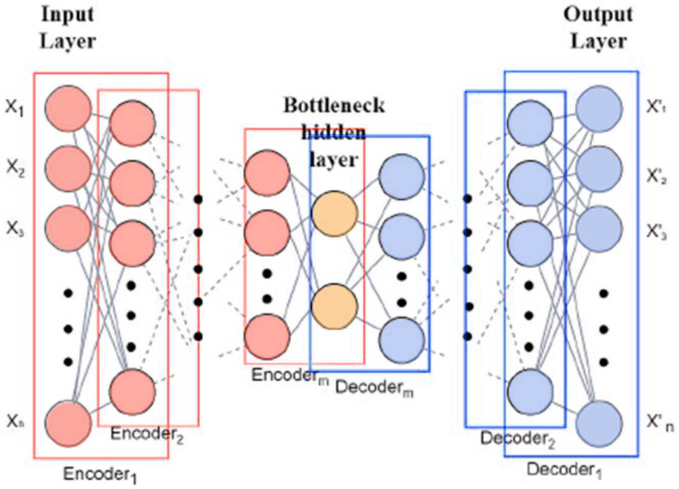

Fig. 3. Deep autoencoder

where F denoted as activation function, x_j is a neuron's input value that was acquired from the layer $l-1$ output of node j. Nodes on the subsequent layer l output from node l as input. This input data is often transformed into an encoding map by autoencoders, which is then further decoded to produce an output layer that represents the input layer's recovered version.

$$X^{'} = D(E(X)) \tag{5}$$

To utilize autoencoders, E and D should be prepared to lessen the contrast among X and X0. An expense capability that works out the mistake between the real and expected values is utilized to contrast the delivered yield X0 with the info X (that is supposed to be created). The model loads are changed during preparing until a decent planning of contributions to yields is delivered to lessen blunder (cost). Deep autoencoders can also use the ANN-specific loss functions. The Mean Squared Error (MSE) is frequently employed. Binary cross-entropy loss is recommended if the input solely contains binary values.

IV. Proposed Method

The research work's methodology is presented in this part. Everyone is aware that the Internet of Things (IoT) is susceptible to a wide range of assaults, including network, software, physical, and privacy-related ones. The new safe IoT framework we provide here allows for the detection of attacks in the IoT environment using a Deep Auto-encoder method. Information extraction from IoT sensor organizations, information preprocessing, information cleaning, highlight extraction, preparing profound learning models, irregularity identification, and effectiveness estimation with accuracy, confusion matrix, recall, and FPR and TPR curves are just a few of the processes that are integrated into the overall framework. This methodology works well for detecting attacks and anomalies in IoT infrastructure. Fig. 4 shows the proposed framework's overall image, which combines numerous distinct sub-processes.

A. Data Source

Our research's main objective is to create a clever, secure, and trustworthy framework for identifying anomalies and assaults in IoT sensor networks. The N-BaIoT dataset, an open-source dataset obtained from Kaggle, is used for experiments with our model. The accuracy, recall, and confusion matrices of the dataset are used to assess the model's efficacy. Our experiments' foundational dataset, N-BaIoT, has likewise been utilized in various examinations on botnet assault location. The vast majority of them utilize parallel or multi-class characterization and classification-based techniques. Our goal is to find an appropriate approach to identifying IoT network traffic anomalies without labelling the raw data first.Network sniffing tools are used to intercept IoT network communications. Some freely accessible tools, such tcp dump and Wireshark, can be used for this. The primary duty is to record network packs, which must subsequently be examined and visualised for analysis. A dataset contains the features that have been retrieved from the network packets.

B. Data Processing

1) Data pre-processing: Data pre-processing is an important stage in learning theories since network data derived from network activity also include these data, which are typically loosely regulated and lead to irrelevant or redundant data values. It cleans up network data by removing unnecessary, distracting, or irrelevant information, which enhances the effectiveness of DE techniques for identifying attack behaviours. The following describes the production, reduction, conversion, and normalisation of features as part of data pre-processing for network data. The preprocessed separated highlights are then used to wipe out repetitive streams, standardize consistent elements, and onehot encode straight out highlights.

2) Data cleaning: We initially locate and eliminate mistakes and duplicate values from the dataset in this process. After that, enter a specific value as "NaN" to replace any missing values. Any machine learning algorithm or model's accuracy and effectiveness can be improved with the aid of this procedure.

3) Feature extraction: We take a conduct depiction of the hosts and conventions that imparted this parcel each time a bundle shows up. The depiction accumulates traffic measurements over various fleeting windows to order the information that was all sent between the source and objective IPs (channel), the source and objective Macintosh addresses, the source and objective TCP/UDP attachments, and the source and objective IPs overall (attachment). Similar arrangement of elements are removed across different time spans. These attributes may be quickly and incrementally calculated, making it easier to identify fraudulent packets in real time.Additionally, despite being general, these qualities can catch specific activities such source IP ridiculing [2], a component of attacks by Mirai. For example, the highlights collected by the Source MACIP, Source IP, and Channel will rapidly uncover a critical irregularity inferable from the concealed conduct coming from the faked IP address when a compromised IoT gadget parodies an IP.

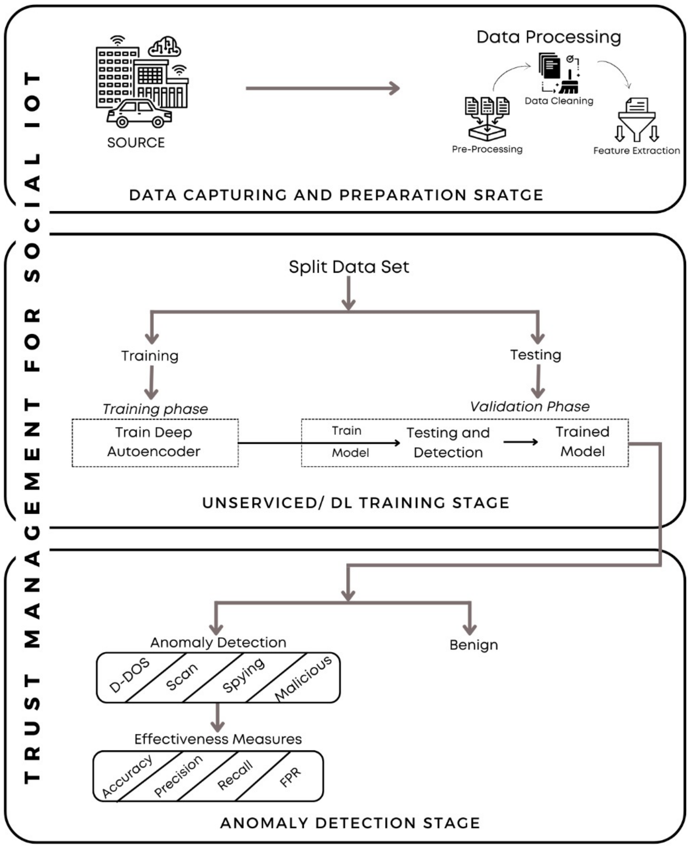

Fig. 4. Proposed Framework for Anomaly Detection

C. Split Dataset

• Training Data: We prepared and streamline a profound autoencoder on 2/3 of the harmless information from every one of the nine IoT gadgets (i.e., the preparation set of every gadget). To record regular organization traffic designs, this was finished.

• Testing Data: Every one of the malevolent information as well as the excess third of harmless information made up every gadget's test information. We utilized the relating trained autoencoder as an anomaly finder on each test set. The detection of anomalies—the hacks perpetrated from each of the aforementioned IoT devices—was successful 100% of the time.

D. Train the Deep Autoencoder

The Harmless and Inconsistency samples are then picked as the objective highlights for the DNN's training on the Train dataset utilizing twofold characterization. A prepared DNN model is made after this stage. We utilize Deep autoencoder as our essential anomaly identifier and exclusively keep a model for each IoT gadget. A neural network that has been trained to adapt its contributions after some upgradation is called an autoencoder. The system will become familiar with the connections between its feedback highlights and important ideas on account of the pressure. An autoencoder will find lasting success at reproducing typical perceptions on the off chance that it is exclusively prepared on harmless cases, yet it will neglect to recuperate unusual perceptions (obscure ideas). We mark the gave perceptions as strange when a sizable reproduction blunder is found.

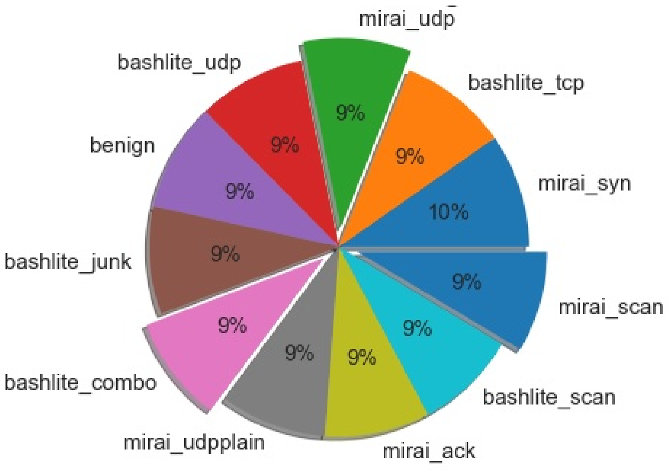

Fig. 5. Distribution of benign and infected devices in dataset

We amplify the genuine positive rate (TPR, distinguishing assaults when they happen) and diminish the bogus positive rate in each preparing model by streamlining its boundaries and hyperparameters (FPR, wrongly stamping harmless information as vindictive). The model learns examples of ordinary movement from two different datasets that are used for training and optimization and only contain benign data.

E. Testing / Detection

The Test dataset is then used to put the trained model to the test, identifying records as either benign or anomalous flows. Benign traffic was permitted to pass through unimpeded if it was anticipated. On the other side, if an anomaly is anticipated, the network administrator is alerted to take further action.

In order to categorise each instance as benign or anomalous, we eventually apply the improved model to feature vectors collected from constantly monitored packets. Then, whether the entire related stream is benign or anomalous is determined by a majority vote on a series of marked occurrences. As a result, if an abnormal stream is found, an alarm might be sent because it might be a sign of malicious activity on an IoT device.

F. Evaluation Metric

It is possible to calculate accuracy using the proposed deep learning framework. Additionally, we can quantify the complexity of our deep learning models, which refers to how many parameters the model contains and how much weight it has when stored to disc, as well as how long the training process takes to obtain good accuracy in terms of time in seconds. The accuracy, precision, false-positive rate, and recall may then be determined for the task of anomaly detection system, which determines the proportion of actual properly recognises instances by model. Eq. (3) demonstrates how to compute recall using (8).

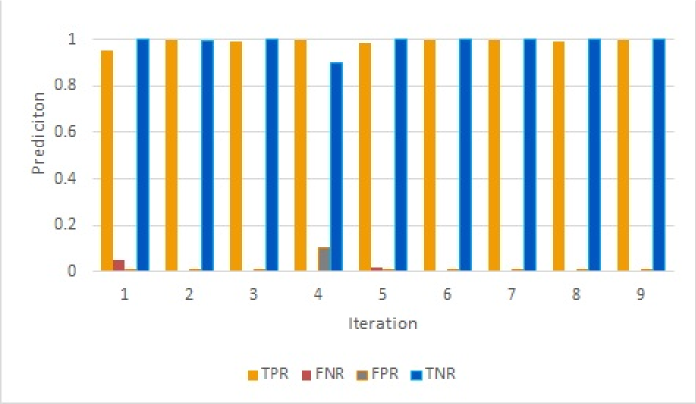

Fig. 6. Prediction probability factor for multi layer perceptron

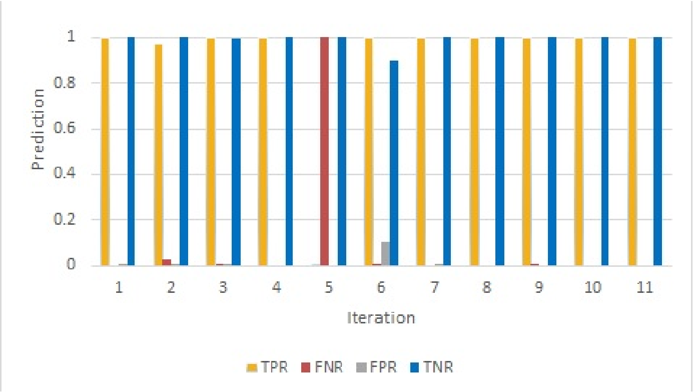

Fig. 7. Prediction probability factor for random forest

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$PR = \frac{TP}{FP + TN} \tag{7}$$

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

Whereas, TP termed as True Positive, FP as False Positive, TN as True Negative,and FN as False Negative, FPR as False Positive Rate, TPR as True Positive Rate, TNR as True Negative Rate, and FNR as False Negative Rate.

V. Results and Discussion

Python was utilised as a platform for implementing the suggested model, and experiments were carried out on an N-BaIoT dataset of an IoT sensor environment. The nine innocuous data sets that we gathered correlate to the nine IoT devices. Fig. 5 shows how different infected and healthy devices are distributed throughout the dataset.

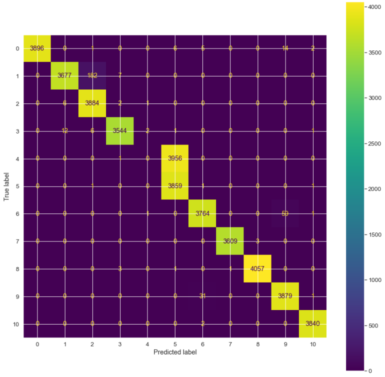

Fig. 8. Confusion matrix of muti layer perceptron

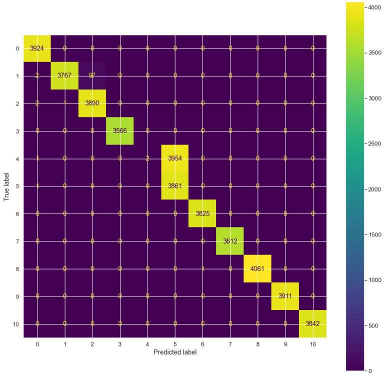

Fig. 9. Confusion matrix of random forest

We offer a comparison of the proposed model, MLP, and RF schemes, with the results for the performance metrics Accuracy, confusion matrix , Precision, F1-Score, Recall and TPR, TNR, FPR, FNR, to confirm the accuracy of the suggested technique. The results of the performance metric taken into account in this study with regard to MLP and RF schemes are summarised in Table III. It is easily shown that the multi-layer perceptron neural network scheme is outperformed by the machine learning-based random forest scheme.

The likelihood of prediction rate for TPR, FPR, TNR, and FNR in MLP and RF systems is shown in Fig. 6 and 7. Misunderstanding Matrix is also constructed for MLP and RF techniques, as shown in Fig. 8 and 9, respectively, to make it simple to spot class-related confusion. It is also referred to as an error matrix and is offered as a table matrix for displaying algorithmic performance and ambiguity in classifier predictions. The performance of the RF technique would not, however, provide the accuracy that is promised as the IoT network's size increases. In light of the massive data set generated by the IoT environment's many IoT devices, the Deep learning-based autoencoder approach is seen as a viable mechanism.

We choose the Window size as 82, Learning rate as 0.01, Optimizer as Adam, activation function as Relu in encoder, and Relu and Sigmoid are utilised in decoder while building the deep learning based autoencoder, referred to as deep autoencoder in this work. The training process's Loss function is the mean square error. Tensorflow was utilised for training.

TABLE III. Perfomance Summary of Multi Layer Perceptron and Random Forest

Scheme	Accuracy	Recall	F1 Macro	F1 Micro	TPR	FNR	FPR	TNR
MLP	89.39%	89.52%	0.86	0.89	0.89386	0.16014	0.01061	0.98939
RF	90.0%	91.0%	0.87627	0.90482	0.90482	0.09518	0.00952	0.99048

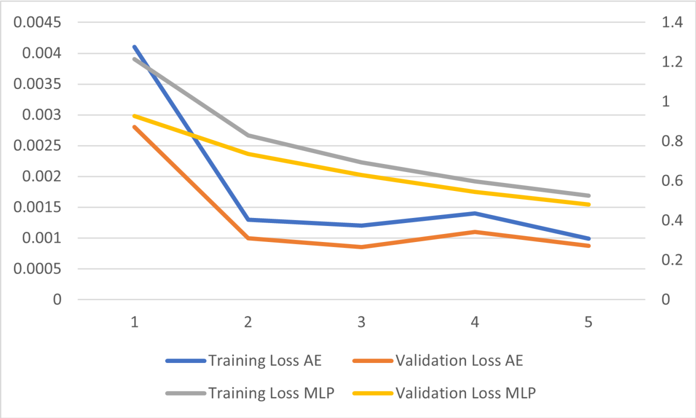

Fig. 10. Loss function of deep autoencoder

The dimension of the input layer for each autoencoder was equal to the number of features in the dataset (i.e. 115). For proper compression of the input layer between encoder and decoder and to reflect its fundamental properties, the autoencoder must effectively execute dimensionality reduction internally.

It is important to appraise the blunder for the model's present status as a component of the improvement technique more than once. To refresh the loads and lower the misfortune on the ensuing assessment, it is important to choose a mistake capability, otherwise called a misfortune capability, that might be utilized to gauge the deficiency of the model. A planning from contributions to yields is advanced by brain network models through models, and the misfortune capability utilized should be suitable for the particular prescient displaying task being tended to, like grouping or relapse. Also, the result layer's design should be reasonable for the chosen misfortune capability. Fig. 10 compares the loss functions for MLP and Deep autoencoder models; when the number of IoT devices is lower, both techniques perform similarly (up to 2). However, when the number of IoT devices grows, the suggested deep autoencoder model outperforms MLP in terms of performance.

The performance of the Deep autoencoder is summarised in Table IV. When compared to MLP and RF techniques, the suggested Deep autoencoder methodology performs better. The outcomes attest to the proposed IoT network solution's superiority. Additionally, it is noted that among the three schemes taken into account in this study, the RF technique is the second-best model and the MLP is the poorest.

TABLE IV. Perfomance Summary of Deep Autoencoder

Node Number	Shape of Data	Detected Anomalies
1	(22154,115)	0.0%
2	(96781, 115)	100.0%
3	(60554, 115)	100.0%
4	(65746, 115)	100.0%
5	(156248, 115)	99.94%
6	(56681, 115)	100.0%

VI. Conclusion

Current internet security measures, such as firewalls and gateways, are ineffective at identifying sophisticated and unidentified assaults in an IoT environment. It is essential to secure this network infrastructure as demand for IoT networks grows. This study explains how AI works to identify assaults and anomalies in the environment of IoT sensors. The detection and classification of IoT botnet attacks using deep learning techniques showed good accuracy. These approaches also function well with a variety of feature counts, and in general, more features do not degrade their efficiency, allowing for the use of all data features in a real-world setting.To recognize benign and irregular traffic, this study proposes a proficient anomaly detection technique in view of deep learning for IoT network design. This system actually gains significant complex examples from IoT network streams. suggested to train and test On the recently made available IoT-Botnet 2020 dataset, a deep autoencoder is tested. Several data processing procedures, including feature extraction, data cleaning, and data pre-processing, are carried out to provide the best outcomes. We construct a number of metrics, including Accuracy, Precision, Recall, Confusion Matrix, and FPR, to assess how well our suggested model performs. A comparison between the proposed model and the current RF and MLP approaches is also done as proof. The ML based RF scheme works with an efficiency of 90.0%, and the neural network based scheme MLP shown the accuracy of 83.39%, while the proposed Deep Learning scheme, deep autoencoder has proved its superiority among the other two methods considered in this study.

The data under consideration in this inquiry is N-BaIoT, which presents 115 aspects of the data samples. A botnet is the type of DDoS attack under consideration. The viruses BASHLITE and Mirai are used to cause network anomalies. For the aforementioned considerations, the results reported in this paper are validated. However, there is still need to investigate the performance of deep autoencoders for various types of datasets and malwares.

References

[1] Harb, H., Mansour, A., Nasser, A., Cruz, E. M., & de la Torre Diez, I. (2020). A sensor-based data analytics for patient monitoring in connected healthcare applications. IEEE Sensors Journal, 21(2), 974-984.

[2] Haider, I., Khan, K. B., Haider, M. A., Saeed, A., & Nisar, K. (2020, November). Automated robotic system for assistance of isolated patients of coronavirus (COVID-19). In 2020 IEEE 23rd International Multitopic Conference (INMIC) (pp. 1-6). IEEE.

[3] Hovav, S., & Tsadikovich, D. (2015). A network flow model for inventory management and distribution of influenza vaccines through a healthcare supply chain. Operations Research for Health Care, 5, 49-62.

[4] Sarkar, N. I., Kuang, A. X. M., Nisar, K., & Amphawan, A. (2014). Performance studies of integrated network scenarios in a hospital environment. International Journal of Information Communication Technologies and Human Development (IJICTHD), 6(1), 35-68.

[5] Mehmood, Y., Ahmad, F., Yaqoob, I., Adnane, A., Imran, M., & Guizani, S. (2017). Internet-of-things-based smart cities: Recent advances and challenges. IEEE Communications Magazine, 55(9), 16-24.

[6] Patel, R., Longini Jr, I. M., & Halloran, M. E. (2005). Finding optimal vaccination strategies for pandemic influenza using genetic algorithms. Journal of theoretical biology, 234(2), 201-212.

[7] Haque, M. R., Tan, S. C., Yusoff, Z., Nisar, K., Lee, C. K., Chowdhry, B. S., ... & Kaspin, R. (2021, January). SDN architecture for UAVs and EVs using satellite: A hypothetical model and new challenges for future. In 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC) (pp. 1-6). IEEE.

[8] Ahmad, F., Ahmad, Z., Kerrache, C. A., Kurugollu, F., Adnane, A., & Barka, E. (2019, April). Blockchain in internet-of-things: Architecture, applications and research directions. In 2019 International conference on computer and information sciences (ICCIS) (pp. 1-6). IEEE.

[9] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150.

[10] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419.

[11] Xiaolong, H., Huiqi, Z., Lunchao, Z., Nazir, S., Jun, D., & Khan, A. S. (2021). Soft computing and decision support system for software process improvement: a systematic literature review. Scientific Programming, 2021.

[12] Haque, M. R., Tan, S. C., Yusoff, Z., Nisar, K., Lee, C. K., Kaspin, R., ... & Memon, S. (2021). Automated controller placement for software-defined networks to resist DDoS attacks. Computers, Materials & Continua.

[13] Apostol, I., Preda, M., Nila, C., & Bica, I. (2021). IoT botnet anomaly detection using unsupervised deep learning. Electronics, 10(16), 1876.

[14] Deekshith Shetty, H. C., Varma, M. J., Navi, S., & Ahmed, M. R. Diving Deep into Deep Learning: History, Evolution, Types and Applications.

[15] Ahmad, Z., Shahid Khan, A., Nisar, K., Haider, I., Hassan, R., Haque, M. R., ... & Rodrigues, J. J. (2021). Anomaly detection using deep neural network for IoT architecture. Applied Sciences, 11(15), 7050.

[16] Darwish, A., Hassanien, A. E., & Das, S. (2020). A survey of swarm and evolutionary computing approaches for deep learning. Artificial intelligence review, 53(3), 1767-1812.

[17] Baig, M. N., Himarish, M. N., Pranaya, Y. C., & Ahmed, M. R. (2018, May). Cognitive architecture based smart homes for smart cities. In 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 461-465). IEEE.

[18] Dinesh, B., Kavya, B., Sivakumar, D., & Ahmed, M. R. (2019, April). Conforming test of blockchain for 5G enabled IoT. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1153-1157). IEEE.

[19] Shen, X., Lin, X., & Zhang, K. (Eds.). (2020). Encyclopedia of Wireless Networks. Cham: Springer International Publishing.

[20] Fahim, M., & Sillitti, A. (2019). Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review. IEEE Access, 7, 81664-81681.

[21] Cook, A. A., Mısırlı, G., & Fan, Z. (2019). Anomaly detection for IoT time-series data: A survey. IEEE Internet of Things Journal, 7(7), 6481-6494.

[22] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., ... & Zhou, Y. (2017). Understanding the mirai botnet. In 26th USENIX security symposium (USENIX Security 17) (pp. 1093-1110).

[23] Czyz, J., Kallitsis, M., Gharaibeh, M., Papadopoulos, C., Bailey, M., & Karir, M. (2014, November). Taming the 800 pound gorilla: The rise and decline of NTP DDoS attacks. In Proceedings of the 2014 Conference on Internet Measurement Conference (pp. 435-448).

[24] Rossow, C. (2014, February). Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In NDSS (pp. 1-15).

[25] Sanatinia, A., & Noubir, G. (2015, June). Onionbots: Subverting privacy infrastructure for cyber attacks. In 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (pp. 69-80). IEEE.

[26] Kwon, J., Lee, J., Lee, H., & Perrig, A. (2016). PsyBoG: A scalable botnet detection method for large-scale DNS traffic. Computer Networks, 97, 48-73.

[27] Shafiq, U., Shahzad, M. K., Anwar, M., Shaheen, Q., Shiraz, M., & Gani, A. (2022). Transfer Learning Auto-Encoder Neural Networks for Anomaly Detection of DDoS Generating IoT Devices. Security and Communication Networks, 2022.

[28] Kang, B. B., Chan-Tin, E., Lee, C. P., Tyra, J., Kang, H. J., Nunnery, C., ... & Kim, Y. (2009, March). Towards complete node enumeration in a peer-to-peer botnet. In Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (pp. 23-34).

[29] Milojicic, D. S., Kalogeraki, V., Lukose, R., Nagaraja, K., Pruyne, J., Richard, B., ... & Xu, Z. (2002). Peer-to-peer computing.

[30] Al-Qatf, M., Lasheng, Y., Al-Habib, M., & Al-Sabahi, K. (2018). Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. Ieee Access, 6, 52843-52856.