Enhancing IoT Security with Deep Stack Encoder using Various Optimizers for Botnet Attack Prediction

Archana Kalidindi[1], Mahesh Babu Arrama[2]

Research Scholar, Department of CSE, Koneru Lakshmaiah Education Foundation, Hyderabad, India[1]
Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Hyderabad, India[2]

***Abstract*—The Internet of Things (IoT) connects different sensors, devices, applications, databases, services, and people, bringing improvements to various aspects of our lives, such as cities, agriculture, finance, and healthcare. However, guaranteeing the safety and confidentiality of IoT data which has become rich in its quality requires careful preparation and awareness. Machine learning techniques are used to predict different types of cyber-attacks, including denial of service (DoS), botnet attacks, malicious operations, unauthorized control, data probing, surveillance, scanning, and incorrect setups. In this study, for improving security of IoT data, a method called Deep Stack Encoder Neural Network to predict botnet attacks by using N-BaIoT bench mark dataset is employed. In this study a new framework is introduced which will improve the performance of prediction rate to 94.5%. To evaluate the performance of this method assessment criteria are adopted like accuracy, precision, recall, and F1 score, comparing it with other models. From the optimizers of Adam, Adagrad and Adadelta, Adam optimizer gave the highest accuracy with relu activation function.**

Keywords*—Internet of things; botnet attacks; neural network methods; N-BaIoT; deep stack encoder; Adam optimizer; Adagrad optimizer; Adadelta optimizer; activation function

I. Introduction

The Internet of Things is developed by wireless sensor networks. Through the Internet of Things, people can connect the physical world with the online world. With the rapid development of integrated circuit technology and wireless communication technology, engineers have been able to create IoT nodes that are very inexpensive and have both signal acquisition, data processing, and wireless communication capabilities [1].

The Internet of Things (IoT) is a network that connects assorted devices to the web through a definite protocol, facilitating data sharing, intelligent identification, tracking, placement, management, and monitoring. While the traditional perception of IoT revolves around a network of physical objects, the internet now encompasses a wide range of devices, including household appliances, smartphones, vehicles, toys, cameras, medical devices, advanced frameworks, individuals, animals, and buildings. These interconnected devices communicate and exchange data according to predetermined protocols.

The IoT's use in the industrial sector is supposed to increase output, efficiency, and security of industrial processes, according to the industry 4.0 vision. In essence, the IoT refers primarily to the effective application of the IoT in industrial operations. The architecture of the IoT can be summed up in four layers. In the industrial sector, the Internet of Things (IoT) architecture comprises of multiple layers: physical, network, middleware, and application. The physical layer encompasses various physical equipment, sensors, mobile and computer devices, as well as other monitoring and automated devices. The network layer encompasses diverse communication networks such as machine-to-machine interfaces, cellular networks, and wireless sensor networks. The middleware layer includes cloud storage, application programming interfaces (APIs), and web services, which facilitate communication between the network layer and the application layer. Finally, at the topmost layer, the application layer enables a wide range of industrial processes and services, including robots, smart factories, smart buildings, smart healthcare, smart vehicles, and more.

IoT systems are made up of interconnected computing devices that can be mechanical, electronic, or any other type of object. For Internet of Things (IoT) systems, it is essential that each device has a unique identifier and the capability to transmit data above a network without relying on human-to-human or human-to-computer interaction. To connect with multiple devices or objects, IoT systems utilize distinctive network address schemes. Unfortunately, a significant number of IoT devices connected to the Internet lack sufficient security measures due to resource constraints, rendering them susceptible to cyberattacks. Yet, the majority of IoT systems run independently across unreliable network connections and the Internet, which exposes the network to cyberattacks. Security concerns need to be resolved as soon as possible given network attacks and cyber threats vs. the bright future of IoT systems.

An IoT network's, Network Intrusion Detection System (NIDS) keeps track of all internet traffic passing through the devices. It acts as a protective barrier that can identify threats and safeguard the network against unauthorized users and malicious attacks. The main defense against network intrusion and other threats in modern computer networks is NIDS. IoT devices are constrained by their physical counterparts' energy consumption, memory capacity, and computational power. Hence, it is nearly difficult to utilize conventional signature-based intrusion detection systems on these devices. Large datasets are frequently needed for signature-based NIDS in order to build reliable detection systems for IoT. The resources of IoT devices must be taken into account when restructuring traditional signature-based NIDS.

In any communication network, the IoT is exposed to various kinds of vulnerabilities and security threats. In particular, security is a critical challenge for the IoT development, as it constitutes an extended version of the conventional unsecured Internet model and combines multiple technologies such as Wireless Sensor Networks (WSNs), optics networks, mobile broadband, and 2G/3G communication networks. Each of the aforementioned technologies is prone to various security risks.[2].

It is anticipated that IoT applications and technologies would advance beyond anything that is conceivable. Unfortunately, IoT technology development is still in its infancy and has not reached its full security protection maturity. IoT software developers' update management issues and non-uniform manufacturing standards are two security challenges faced by IoT systems. Critical challenges include the physical management of security concerns and users' ignorance as a result of their ignorance of security issues related to IoT devices. The network and surroundings of IoT systems must also be protected, in addition to using encryption techniques to secure data transmission.

However, the nature of the resource limitations prevents the use of conventional network security mechanisms in IoT systems. Due to the IoT system applications' quick development and widespread adoption, several network attacks have also surfaced. The number of assaults will increase as IoT use cases develop. Being aware of the substantial increase in cyber-threats within the IoT system significantly mitigates the probability of network security breaches and data compromises.

Some examples of the most prevalent attacks launched against IoT systems include:

A. *Denial of Service (DoS)*

Due to enormous cyberattacks IoT systems or network resources become unreachable to the intended authorized users. The purpose of these attacks is to temporarily or permanently interrupt the services provided by a host IoT system.

B. *Distributed Denial-of-Service (DDoS)*

A distributed DDoS attack is a malicious network attack that interrupts systematic traffic and network services. It involves overwhelming the target or neighboring infrastructure with a disproportionate volume of network traffic. DDoS attacks are effective when attackers exploit various compromised systems to produce a huge volume of traffic in the network. IoT systems or other devices which are the part of the network can also be targeted with these attacks.

C. *Marai Botnet Attack*

Cybercriminals employ the software known as Mirai to turn networked devices into remotely controlled robots in a catholic scale network as a part of botnet. It primarily targets internet consumer electronics, including IP cameras and routers for the house. Mirai was frequently used as an initiator in attacks like DoS/DDoS.

D. *Sybil Attack*

Peer-to-peer networks are susceptible to Sybil attacks. A Sybil attack alters the identity of the IoT device to generate numerous anonymous identities and use an excessive amount of power. It was given that name in honor of Sybil, who wrote the book Sybil, in which a lady coping with dissociative identity disorder. An IoT device in a network that uses several identities frequently compromises reputation systems' allowed network access. Attacks using Sybil take use of this vulnerability in the IoT system network to launch initial attacks.

Since 2007 AI-based threats have been arisen as a significant trouble to the Internet of Things (IoT). These attacks, driven by artificial intelligence, present a greater danger compared to traditional human-focused attacks. Cybercriminals now leverage AI-powered tools that are faster, scalable, and more efficient, posing a serious challenge to the IoT ecosystem. The nature of AI-based assaults, with their increased volume, automation, and customization, makes them difficult to counter, despite sharing certain characteristics and strategies with traditional IoT hazards.

Further down, reader can see the literature survey which talks about the previous works and findings, followed by the proposed work and methodology, which covers about the information regarding the dataset, clean up and pre-processing techniques of the data, modules and tools used in the proposed work and libraries used for implementation of the proposed work, in succession there are algorithm which explains the detailed flow of the project from pre-processing to results and system architecture explaining the structure of the proposed work.

II. Literature Review

The proliferation of the Internet of Things (IoT) has observed significant progress, making it vulnerable to cyber-attacks targeting IoT devices. Safeguarding these devices to give security has become a crucial priority in order to mitigate potential risks. Among the various types of attacks, botnet attacks pose a severe and pervasive threat to IoT devices. One vulnerability lies in stationary IoT devices, as they often lack the necessary memory and computational capacity required for robust security measures. Moreover, several current systems are dedicated to enhancing security by identifying unfamiliar patterns within IoT networks [3].

The fundamental concept behind the Internet of Things (IoT) is to unite smart devices to the web, enabling seamless communication between physical objects and various entities like servers and mobile devices. The IoT has made its way into every sphere of life, spanning homes, industries, healthcare, automotive, and sensors. Consequently, the proliferation of vulnerabilities within IoT security poses severe risks to user safety and property [4].

The Internet of Things (IoT) industry flourishes; it has a significant rise in the diversity and abundance of IoT devices. These devices find widespread application in areas such as smart homes, wearable technology, manufacturing, automotive, and healthcare, and other domains related to daily life. However, this rapid expansion also leads to a continuous emergence of security vulnerabilities in IoT devices. The escalating number of security vulnerabilities poses substantial risks to the privacy and property of users [5].

We have in-depth analysis of detection and prevention methodologies for various security attacks aimed at IoT systems. It is specifically aimed at software developers, researchers, and professionals working in the field of Internet of Things, who desire a comprehensive understanding of the strategies employed to detect and mitigate these attacks. Each item in the list is accompanied by a concise description and references that readers can refer for more detailed information [6].

The Industrial Internet of Things (IIoT) encompasses a wide range of elements, including sensors, machinery, industrial applications, databases, services, and workforce, which collectively contribute to various aspects of lives such as smarter cities, agriculture, and e-healthcare. While IIoT and consumer IoT share certain similarities, distinct cybersecurity measures are implemented for each network. Unlike consumer IoT, which is typically utilized by individual users for specific purposes, IIoT solutions are often integrated into larger operational systems. [7].

IoT-enabled devices have found applications in both industrial and commercial sectors, offering businesses a competitive edge over their rivals. However, the widespread use of interconnected smart devices has led to heightened concerns regarding privacy and data breaches. These concerns have disrupted workflow, activities, and network services within enterprises. To safeguard their organizational assets and ensure uninterrupted services, professionals must proactively address these risks by implementing comprehensive security protocols and policies [8].

One area that has received limited attention in previous literature is the vulnerability of routing protocol for low power and lossy networks to attacks. To address this issue, the author of this study proposed an artificial neural network (ANN) model for detecting decreasing rank attacks. The results showcased an impressive accuracy rate exceeding 97% and demonstrated strong performance across various tests conducted on the held-out testing dataset. These findings indicate the model's efficacy in terms of accuracy, precision, detection probabilities, false-positive rate, false-negative rate, and other relevant metrics [9].

The scientific community has shown considerable interest in the Internet of Things (IoT). The potential compromise of these devices by malicious individuals not only jeopardizes privacy but also poses significant risks to critical assets. Consequently, the detection and prevention of unique attacks within the IoT ecosystem are of utmost importance. In this study, the author introduces a novel threat detection system that integrates development and operations frameworks. In the initial phase, data from each application is processed by incorporating statistical and higher-order statistical features alongside the existing ones [10].

The integration of the Internet into corporate processes through IoT platforms becomes more prevalent, the need for stable and efficient connections becomes increasingly important. The authors of the article introduce a comprehensive automated intrusion detection system that focuses on enhancing Fog security and addressing cyber-attacks. The proposed model utilizes multi-layered recurrent neural networks that are specifically designed for deployment in Fog computing environments, which are situated in close proximity to end-users and IoT devices. Given that intrusion detection systems are among the key remedies employed for IoT security, it is common to adopt multiple strategies simultaneously. RNN and other neural networks can be effectively employed to analyze data and provide protection against cyber threats, offering layered defense mechanisms [11].

Employing machine learning within an IoT gateway helps protect the system in order to address the issues of securing IoT devices. They examine the use of Artificial Neural Networks in a gateway to detect anomalies in data transmitted from edge devices and are persuaded that this method can improve IoT system security. Security has been regarded as one of the weaker aspects in IoT during its growth. There are various hurdles to implementing security inside an IoT network, including system heterogeneity and the sheer number of devices that must be addressed [12].

All information processing systems now include a fundamental component for the detection of cyberattacks, and once an attack is identified, it might be possible to stop it or lessen its effects. In this study, the focus is on developing a straightforward detector to identify specific Botnet attacks on IoT systems. The proposed approach involves utilizing a learning recurrent random neural network (RNN), which offers advantages in terms of its compact 12-neuron recurrent architecture and low computational requirements, making it well-suited for edge devices. The RNN is trained offline using a simplified gradient descent technique, resulting in high detection rates of approximately 96% while maintaining minimal false alarm rates [13].

Security plays a critical role in nearly implemented or ongoing IoT applications. The widespread adoption of IoT is rapidly expanding and infiltrating various industries. While current networking technologies offer support for many IoT applications, certain applications demand more robust security measures from the underlying technologies they rely on. Looking ahead, IoT devices will not only be connected to the internet and local devices but will also have the capability to directly communicate with other devices across the internet [14].

The present era is characterized by an extensive deployment of IoT systems that generate vast amounts of data, and the detection of anomalies is a crucial aspect of every such system. These anomalies may indicate resource depletion in an industrial environment, unforeseen issues at an aerospace platform, or unusual performance of medical devices, among others. Hence, the ability to identify anomalies can have any monitoring system's overall performance is significantly impacted. The dataset in this context includes several forms of threats, such as DoS/DDoS, Botnet, Brute Force, Web Attack, Infiltration, and Port Scan, that could potentially cause an IoT system to fail[15].

The number of Internet-connected devices, such as cameras, embedded machines, sensors, and many others that comprise the IoT, is rapidly increasing. By 2025, as projected by the International Data Corporation (IDC), the number of interconnected IoT devices is estimated to reach 41.6 billion. DL-based security mechanisms are heterogeneity tolerant since they can learn diverse features from unstructured data on their own. They can also be utilized to distinguish novel mutated threats from their older incarnations; thus, the security mechanism does not necessitate a patch on IoT devices on a regular basis [16].

The applications of IoT are expansive and continuously expanding, covering a wide range of areas such as public security, infrastructure development, connected healthcare, smart homes, cities, grids, and wearables. However, with such widespread use comes the risk of various attacks, including those aimed at denying service or taking control of the network. Among these, DDoS attacks pose a significant threat to IoT systems, as they involve many attackers from different locations overwhelming the network. To combat this, the author suggests using SDN and recurrent neural networks for DDoS detection and IoT security [17].

As the usage of IoT devices becomes increasingly widespread, network attacks have grown in frequency and severity. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks have emerged as common types of threats targeting IoT networks. Traditional security measures like firewalls and intrusion detection systems are insufficient when it comes to detecting complex DoS and DDoS attacks. This is because they rely on static predefined rules to differentiate between normal and malicious network traffic [18].

The Internet has evolved from being a useful research tool for academic institutions to becoming an essential utility, comparable to gas, water, and electricity. However, as with any valuable resource, there is a risk of crime intended to exploit the technology illegally or to impede others from using it. The interconnectedness of the Internet makes it vulnerable to attacks from anywhere in the world, making cybersecurity a crucial concern. According to the latest survey conducted in 2015, security breaches are increasing [19].

This passage delves into an in-depth exploratory study that examines the obstacles associated with integrating these technologies into a cohesive system. The integration is affected by various challenges including security, scalability, accountability, and issues related to communication trust. The successful and effective integration of these technologies can accelerate the digital transformation of market, companies and the development of new business models [20].

The advent of the Internet of Things (IoT) has transformed the traditional way of living by introducing a sophisticated way of life. IoT has brought about numerous innovations such as smart homes, smart cities, pollution control, energy conservation, smart transportation, and smart industries. Numerous important studies and research have been carried out to develop technology through IoT [21].

Copious IoT devices are presently available for use, many of which are extensively used in various services and are vulnerable to cyber-attacks. Cyber-attacks targeting IoT devices do not only affect the devices themselves. Since IoT devices are usually connected to other systems and appliances, they become entry points for hackers to gain access to anything connecting them [22].

The Internet of Things (IoT) has given rise to world of limitless opportunities for applications across many facets of society, but it also comes with several difficulties. Security and privacy are two such issues. To address this issue, incorporating security measures into the hardware of IoT devices beyond standard procedures is a potential solution [23]. A few examples of the devices are laptops, cell phones, tablets, washing machines, etc. IOT is a vast network of linked "things." The devices each have a microchip that connects them all. These microchips monitor their environment and report back to both humans and the network. The best feature of IOT is that every physical object may connect with one another and is reachable over the internet. Many devices are linked to the internet as a result of cheap internet access [24].

A method of identifying the neuron's structure and the optimal activation function of stacked autoencoders has been proposed for dimension reduction to minimize mean square error loss. A total of eight different neuronal structures of auto encoders and six activation functions are used to accomplish this. As a result, the optimal structure is 68-50-30-58-60 when viewed from the perspective of the mean squared loss function. As far as computational time and classification metric (97.4%) are concerned, the ELU is with negligible difference in the best activation function. It has been stated in [25] that this study will assist the defenders in selecting the activation method. In [26] it is recommended that activation and loss functions that may be useful to defenders. By using the CICIDS 2017 dataset, the effect of these functions is evaluated with an SVM-RBF classifier.

In [27] a model has been advocated by using semi-supervised Deep Learning, specifically Semi-supervised GAN (SGAN), for detecting botnet attacks on the N-BaIoT benchmark dataset is interesting. It appears that the approach has achieved high accuracy for binary classification (99.89%) and a decent accuracy for multiclass classification (59%). Semi-supervised learning techniques can be useful when labeled data is limited or unavailable. By leveraging both labeled and unlabeled data, semi-supervised models can learn from the available labeled data while utilizing the unlabeled data to improve the model's performance.

III. Proposed Work

In this work, the main concentration is on increasing the accuracy even for multi class classification by using autoencoders and also by considering confidentiality, availability, integrity, and privacy as they are more specific security needs, which are frequently referred to as security attributes. The technology tries to reduce latency and improve reliability while data is transmitted across the network. In order to identify attacked data in the IoT context, this system makes use of the reputation model. In this proposed model Adam optimizer and Average subtraction-based optimizer are used which increases accuracy as compared to existing models. To make sure in terms of the security in the Internet of Things, it is crucial to accurately identify the interconnected devices. This involves employing a technology that can automate three key functions related to IoT security, specifically for device identification and discovery. IoT devices on the network are automatically and continuously detected, profiled, and categorized. Also it keeps a running list of the gadgets.

A. *Data-Set Collection*

For this study, the dataset which is used contains 10 lakhs of rows and 115 columns. This dataset was taken from [28]. It has many rows and columns and tried to include all the types of possible botnet attacks. The dataset which is used is N-BaIoT. The N-BaIoT dataset is a state-of-the-art and exceedingly refined assemblage of data that holds the capacity to revolutionize research within the realm of Internet of Things (IoT).

Table I gives brief information about IoT devices which have been used in two different botnet with their model names in the dataset considered. This dataset encompasses an extensive array of sensor readings and significant metrics, delivering a comprehensive and meticulous overview of the condition and conduct of IoT devices in authentic, real-life surroundings. The ongoing study undertakes an exhaustive exploration of the intricate and exceptionally advanced N-BaIoT dataset, encompassing an astonishing 7,062,606 records of network traffic, comprising both malevolent and benign activities.

The Table II depicts different IoT botnets which consists of various types of attacks and have been collected from a simulated organizational context. Last two columns of the specified dataset are the output columns which tells us whether the IOT devices are attacked or not and the category of the attack. The proposed model is checking whether the IOT devices undergone by botnet attacks or not and even specify the type of attack.

The graphs which are depicted for Marai and Bashlite bonnets (Fig. 1 and 2 respectively) consist of various types of attacks in individual botnets. These attacks are harmful as the complete network will be in the control of botmaster, the attacks which have been discussed, occur may be due to the sensitivity of IoT devices in the network [29]. The pursued dataset contains two types of Botnets and each one of it contains five different malwares and number of each has been depicted in the figures.

TABLE I. DEVICES IN N-BAIOT DATASET WITH MODEL NAMES

Types of devices with their model names
Danmini_Doorbell
Ecobee_Thermostat
Ennio_Doorbell
Philips_B120N10_Baby_Monitor
Provision_PT_737E_Security_Camera
Provision_PT_838_Security_Camera
Samsung_SNH_1011_N_Webcam
SimpleHome_XCS7_1002_WHT_Security_Camera
SimpleHome_XCS7_1003_WHT_Security_Camera

TABLE II. DIFFERENT BOTNETS AND TYPES OF ATTACKS IN N-BAIOT DATASET

IoT Botnets	**Types of Attacks**
Mirai	ACK
	Scan
	Syn
	UCP
	UDP Plain
Bashlite	Combo
	Junk
	Scan
	TCP
	UDP

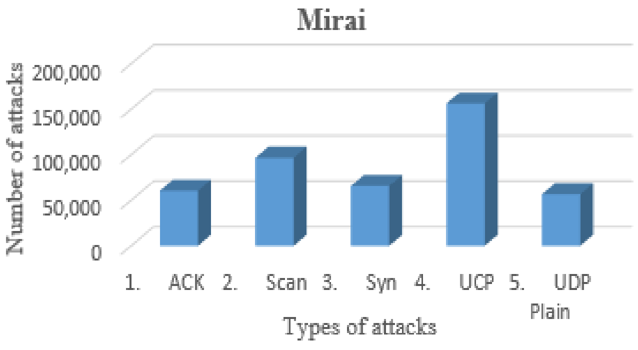

Fig. 1. Distribution of different attacks in Marai botnet.

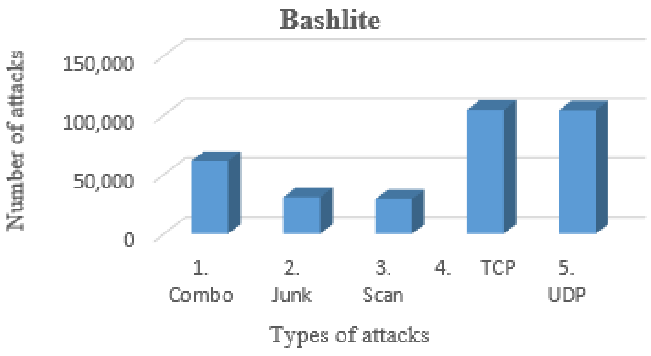

Fig. 2. Distribution of different attacks in Bashlite botnet.

B. Normalization

Normalization is an adequate preprocessing technique which has various normalization methods as well. The study [30] uses min max normalization for intrusion detection in the network to identify malwares; however Z-sore normalization is one good option. Z-score normalization, also known as standardization, is a method used to transform the values of a feature in a dataset to have a mean (μ) of 0 and a standard deviation (σ) of 1. The transformation is performed while from each value is subtracted from the mean of the feature, and then dividing by the standard deviation. The transformed feature is then referred to as a standard score or a Z-score (eq 1). This normalization method is commonly used in machine learning and data preprocessing to ensure that all features are on a similar scale and to reduce the impact of outliers. Hence, the normalization technique is used to normalize the data given in the dataset and also by standardizing the features, it can also improve the numerical stability and convergence speed of some machine learning algorithms. Comparing to the other normalization techniques named min-max normalization, long scaling and clipping, and BCNF. Z-Score normalization gave highest accuracy.

$$Z - score(X) = \frac{(X-\mu)}{\sigma} \qquad (1)$$

C. Feature Selection using Information Gain

Information gain is a feature selection method used in machine learning to rank the importance of features based on the reduction of entropy in the data. In decision tree learning, information gain is used as a criterion for splitting the data based on the features. The entropy of a set of samples represents the amount of uncertainty or randomness in the data. By selecting features with high information gain, the entropy of the data is reduced, leading to a more predictable and accurate model. The idea is to select features that provide the most information about the target variable, by measuring the reduction in entropy after splitting the data based on each feature. Information gain for feature selection has been used which is a simple and effective feature selection method that can be used in various machine learning algorithms, especially decision trees and decision tree-based ensemble methods. Let F be the set of selected features then,

$$F = argmax(InfoGain(X_i, Y)): X_i \in X \qquad (2)$$

D. Data Processing

For data processing One Hot Encoder is used, which is a data transformation technique used in machine learning and data analysis. The process involves transforming categorical variables into a format that is compatible with machine learning algorithms. In one hot encoding, each unique categorical value in a column is converted into a binary vector of 0s and 1s. For example, if a categorical column has three possible values "A", "B", and "C", the one hot encoding process would convert this column into three binary columns: one for "A", one for "B", and one for "C". If a row had the value "B" in the original column, then the "B" column would have a 1 in that row and the other two columns would have 0s. So for converting strings into numerical values One Hot Encoder was used for machine learning algorithms.

IV. Methodology

The architecture represented in the diagram (Fig. 3) is a deep stack encoder, consisting of numerous layers that are arranged on top of each other. This design allows the model to learn hierarchical representations by gradually extracting complex features from the input data. Before feeding the data into the encoder, a feature selection process is performed using information gain. Out of the original 115 features, 58 features are nominated based on their relevance and prominence to the task at hand. This choice helps to reduce the dimensionality of the input and emphasis on the most informative features. To optimize the model's parameters and improve the efficiency of training, different optimizers are employed. Precisely, the optimizers used in this work include Adam, Adagrad, and Adadelta. These optimizers regulate the weights and biases of the model throughout training, with the goal of lessening the loss function and taming the model's performance.

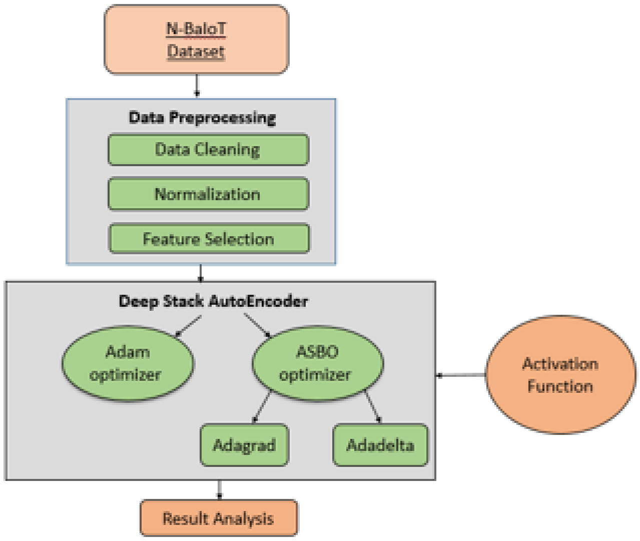

Fig. 3. Workflow diagram of proposed methodology.

The special choice of activation function is one more important factor in the model design. In this case, Rectified Linear Units (ReLU) are used as the activation function for both the input and hidden layers. ReLU activation is known for its ease and effectiveness in supporting the model to learn intricate nonlinear relationships in the data. For the output layer, the softmax activation function is employed which is commonly used in multi-class classification tasks as it converts the output values into a probability distribution over diverse classes, enabling the elucidation of the final predictions. By deploying these design choices, such as feature selection, diverse optimizers, and apposite activation functions, the model targets to attain superior performance and acquire better-quality outcomes for the specified task.

A. Deep Stack Auto Encoder

Deep Stack Encoder is also known as stacked autoencoders, which are used in unsupervised learning. Stacked autoencoders consist of multiple layers of autoencoder neural networks. An autoencoder is a type of neural network that is trained to encode and then decode input data, such that the output is as close as possible to the original input. By stacking multiple layers of autoencoders on top of each other, each layer learns to encode the input data in a more abstract and compressed way, with the final output being a low-dimensional representation of the input data. So Deep Stack Auto Encoder is used to encode and then decode the data. The model is having two types of optimizers, Adam Optimizer and Average Subtraction based Optimizer.

B. Neural Network Using Adam Optimizer

Adam (Adaptive Moment Estimation) is an optimization algorithm widely used for training deep learning models, specifically neural networks. It combines the strengths of two popular optimization algorithms, namely RMSProp and Momentum. The algorithm computes a weighted average of previous gradients and squared gradients to adapt the learning rate on a per-parameter basis, as depicted in equation (2). This capability allows the algorithm to assign different learning rates to individual parameters, resulting in faster convergence and often superior performance compared to other optimization algorithms. The application of the Adam optimizer is done in present study, specifically focusing on two parameters, Y_1 and Y_2. The present model comprises one input layer, three hidden layers, and one output layer. To utilize the Adam optimizer effectively, Tensor Flow and Keras libraries are imported . Finally, the network is compiled to prepare it for further processing.

$$\theta_t = \theta_{t-1} - \alpha * m_t/(sqrt(v_t) + epsilon) \quad (3)$$

where, θ_t is the parameter vector, α is the learning rate, m_t & v_t are the first and second momentum updates.

C. Neural Network using Average Subtraction Based Optimizer

Average subtraction based optimizers are a class of optimization algorithms for training neural networks. They are called average subtraction based because they subtract the moving average of the gradient from the current gradient in order to update the weights. This helps to reduce the variance of the gradient and stabilize the training process.

One of the most well-known average subtraction based optimizers is the Adagrad optimizer. Adagrad updates the learning rate for each weight in the network based on the historical gradient, with a larger learning rate for weights (shown in Eq. 3) with a smaller historical gradient and a smaller learning at optimizer to adapt to the characteristics of each weight and reduces the risk of oscillations or stagnation during training. Another example of average subtraction based optimizers is the Adadelta optimizer, which extends the idea of Adagrad by using the average of the squared gradient instead of the gradient itself. Adadelta also includes a decay factor to reduce the impact of historical gradients over time shown in Eq. 4. Here average subtraction for two parameters Y1 and Y2 was done. In this model, there is a input layer, a hidden layers and a output layer. And for this adadelta optimizer TensorFlow, keras were imported.

$$\beta_t = \beta/sqrt(G_t + epsilon) \quad (4)$$

where β_t is the initial learning rate, epsilon is a small constant to prevent division by zero and G_t is the gradient.

$$E[\delta_w]_t = \rho * E[\delta_w]_t - 1 + (1 - \rho) * \delta_{w_t^2} \quad (5)$$

where ρ is a decay rate that controls the contribution of past gradients to the moving average.

D. Pandas

It is software package for the Python programming language that is used to manage and evaluate data. It is used in particular for huge calculations or bigger data; it has additionally Numpy in it. To perform operations on data files such as csv, pandas library is used. The pd.read_csv( ) feature is utilized to import and analyse the data stored in a csv file. Additionally, to make accessing data easier, names are given to each column and store them in an index list.

V. Algorithm

Algorithm: Deep Stack Encoder with Feature Selection, Adam, Adagrad, and Adadelta Optimization.

1. Load the dataset (D) and store as matrix X with dimensions (n_samples, n_features) and vector Y with dimensions (n_samples,).
2. Normalize the input features of X using Z-Score normalization.

   $$\boldsymbol{Z - score(X)} = \frac{(X - \mu)}{\sigma}$$

3. Perform feature (F) selection using Information Gain,

   $$\boldsymbol{F = argmax}(InfoGain(X_i, Y)) : X_i \in X$$

4. Build three neural network models using Adam, AdaGrad, and Adadelta optimizers, and store them asM_1, M_2, and M_3 respectively.
5. for M_i in Models do
6. Train dataset $\boldsymbol{D_{train}}$ using the selected features F as input features and $\boldsymbol{Y_{train}}$ as output labels.
7. Predict the output labels for the test dataset $\boldsymbol{D_{test}}$ using the trained model and the selected features F as input features and store the predicted output labels as $\boldsymbol{Y_{pred(i)}}$.
8. Calculate the performance metrics for each model $\boldsymbol{M_i}$ using the true output labels $\boldsymbol{Y_{true}}$ and predicted output labels $\boldsymbol{Y_{pred}}$.
9. End
10. Output the performance metrics for all three models, METRICS ($\boldsymbol{M_1}$), METRICS ($\boldsymbol{M_2}$), and METRICS ($\boldsymbol{M_3}$).

Once the dataset have been loaded that consists of 115 columns where it is very huge and for which there is a need to decrease the number of columns. This can be achieved through feature selection by selecting top features by adapting information gain. Then normalize the data points by inheriting Z-score normalization and then construct three neural network models M1, M2 and M3 with three different optimizers Adam, Adagrad and Adadelta. Split the pre-processed dataset into two major division in ratio of 7:3 for training and testing respectively and predict the output labels. Finally calculate the efficiencies by considering performance metrics.

VI. System Architecture

System architecture is a pictorial depiction of all the components that come at a place to procedure the complete system. The architecture of the model is shown in Fig. 4, which also lists all the plans, tools, processes, and other components. Using the provided data set, leveraging the given dataset, two optimization techniques are employed, namely Adam and average subtraction-based optimizers, to enhance the dataset's performance and determine the accuracy of the model. By utilizing these optimizers, the aim is to fine-tune the dataset and achieve improved results. The proposed approach follows a sequential flow for attack prediction. It begins with the user loading the dataset, followed by data preprocessing to prepare the data for analysis. Feature selection techniques, such as information gain, are applied to identify 58 relevant features from the original set. When a user provides input, such as network logs or suspicious activity patterns, the deployed model processes the data and generates predictions regarding the likelihood or classification of an attack. This approach combines dataset loading, data preprocessing, feature selection, model training, deployment, and user input to expedite accurate attack predictions.

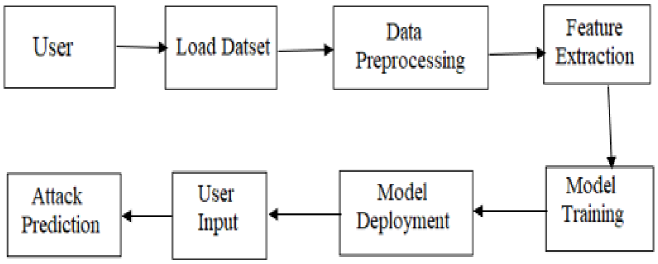

Fig. 4. System architecture.

VII. Experimental Results

Three different types of optimizers are used in the present neural network models, the results are as follows:

Accuracy, recall, precision, and F1-score metrics were considered to test the system for detection of botnet attacks. The equations are defined as follows:

Accuracy: It is the proportion of correct predictions made by the model out of all predictions. It is usually expressed as a percentage.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

Where TP: True Positives, TN: True Negatives, FP: False Positives, FN: False Negatives.

Precision: It measures how many of the predicted positive instances are positive. It is a useful metric when the cost of false positives is high.

$$\mathrm{Precision} = \mathrm{TP} / (\mathrm{TP} + \mathrm{FP})$$

Recall: It measures the ability of a model to identify all positive samples correctly. A high recall indicates that the model is good at identifying positive samples, while a low recall suggests that the model is missing some of the positive samples.

$$Recall = TP / (TP + FN)$$

F1 Score: It is a metric used in binary classification problems, which is the harmonic mean of precision and recall. It takes both precision and recall into account to provide a balanced evaluation of the model's performance.

$$F1\ Score = 2 * (Precision * Recall)/(Precision + Recall)$$

TABLE III. Comparision Table for Model Evaluation with Matrics

	Existing Model	*Feed Forward Adam*	*Adagrad*	*Adadelta*
Accuracy	90.88	94.5	89.9	87.1
Precision	93	96.3	87	84.4
F1 Score	88	96.2	86.3	81.6
Recall	91	97.5	86.5	82.3

In the above table, Table III the performance of an existing model is assessed along with three optimization algorithms, to be precise Adam, Adagrad, and Adadelta, based on accuracy, precision, F1 score and recall metrics. The observation from the comparative study concluded that Adam outstripped the others in terms of accuracy, precision, F1 score and recall. It has been accomplished well with respect to overall performance and to facilitate the rightly classified instances. It is precisely vital to make a note that these results are definite to the considered dataset and may differ according to the type of the data and the job at hand.

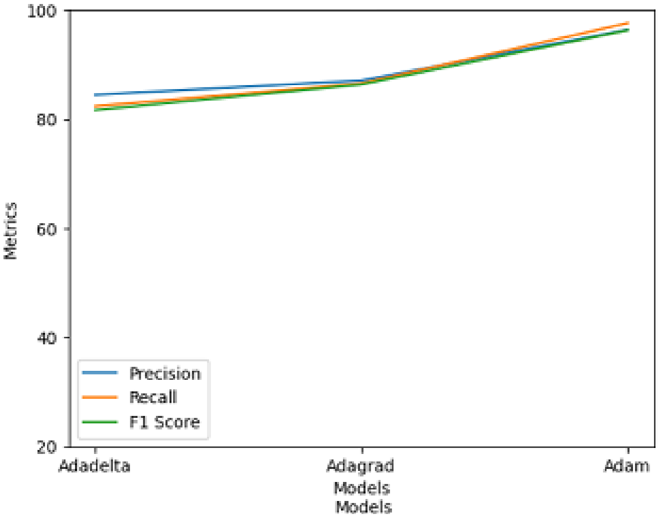

Fig. 5. Comparative graph for precision, recall, f1 score

In credit to adaptive learning rate of Adam optimizer projects higher results where it regulates the learning rate for each parameter during the training session. This adaptive nature guarantees that the model defined converges efficiently without overrunning or getting stuck in local optima. Fig. 5 is the comparative graph which depicts Adam optimizer grander performance in terms of precision, recall, and F1 score when compared to Adagrad and Adadelta.

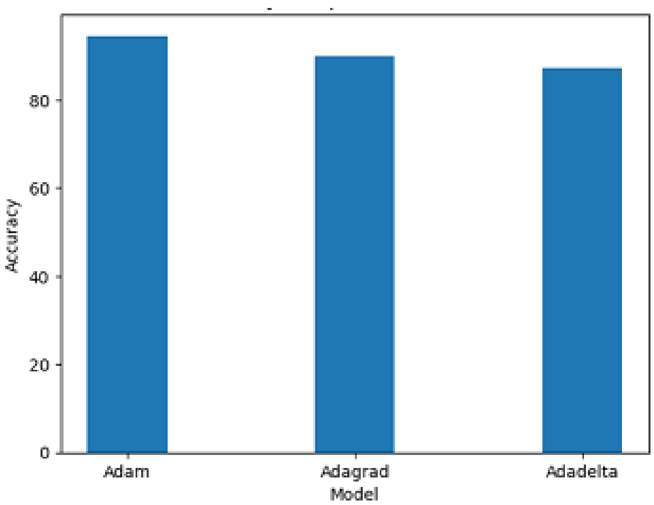

Fig. 6. Comparative graph for accuracy (%).

The major aim of this comparative graph which has been portrayed in Fig. 6 is used to appraise the performance of three optimizers-Adam, Adagrad and Adadelta. The fallouts of the analysis exhibits the Adam optimizer efficiency when compared with other two. The visual graph presented embodies x-axis with optimizer names and y-axis with accuracies with a bar graph. The bar for Adam optimizer stood above all the other optimizers signifying its efficiency.

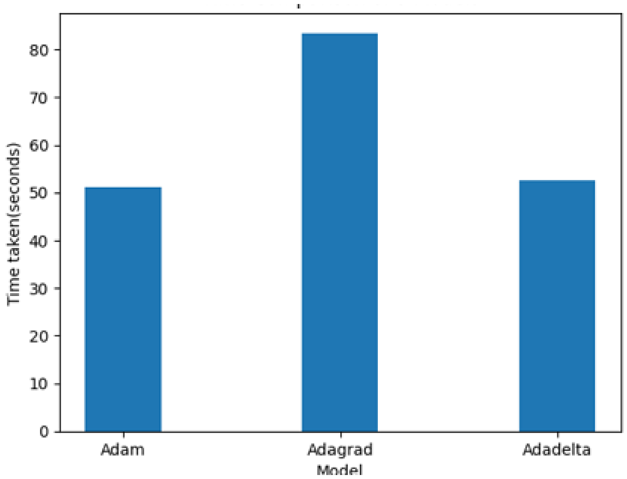

Fig. 7. Comparative graph for time complexity.

The time complexity of a neural network model is influenced by various factors, including the number of layers, the number of neurons in each layer, the type of activation function utilized, the number of training epochs, and the size of the input data. Typically, the time complexity of a neural network model can be expressed as O(N^3), where N represents the number of neurons in the largest layer. The time taken to get the results for Adam, Adagrad, Adadelta are 53.63 sec, 68.52 sec and 65.32 sec (see Fig. 7). When there is a need for quick and superior results, commissioning the Adam optimizer is extremely recommended when it is related with other optimizers mentioned.

VIII. Conclusion and Future Enhancement

In this work, the use of different kinds of optimizers named Adam optimizer, average subtraction based optimizer which contains Adagrad optimizer and Adadelta optimizer was done. These are the parts of deep stack encoder. Adam optimizer gave the accuracy of 94.56, Adagrad gave the accuracy of 89.95, and Adadelta gave the accuracy of 87.17. From the experimental results we conclude that Adam optimizer is the most accurate optimizer and less time taking. For future intensifications, we can change the number of hidden layers and number of neurons in input, output and hidden layers to increase accuracy further. As in here, there is only use of one input, one hidden and one output layers. The change in the number of layers can be multiple combinations which will bring significant difference in the results. The number of activation functions can change the future results.

References

[1] D. Li, L. Deng, M. Lee, and H. Wang, “IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning," Int. J. Inf. Manage., vol. 49, pp. 533-545, Dec 2019.

[2] P. I. R. Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, "Securing the Internet of Things: Challenges, threats and solutions," Internet Things, vol. 5, pp. 41-70, Dec 2019.

[3] H. Alkahtani, and T.H. Aldhyani, "Botnet attack detection by using CNN-LSTM model for Internet of Things applications," Secur. Commun. Netw. , pp. 1-23, Sep 2021.

[4] K. Ali, and S. Askar, "Security Issues and vulnerability of IoT devices," DaInt. j. sci. bus., vol. 5(3), pp.101-115, Feb 2021.

[5] M. Yu, J. Zhuge, M.Cao, Z. Shi, and L. Jiang, "A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices," Future Internet, vol. 12(2), p. 27, Jan 2020.

[6] M. Shafiq, Z. Gu, O.Cheikhrouhou, W.Alhakami, and H.Hamam,"The rise of “Internet of Things”: review and open research issues related to detection and prevention of IoT-based security attacks," Wirel. Commun. Mob. Comput., pp.1-12, Aug 2022.

[7] S. Latif, Z. Zou, Z. Idrees, and J. Ahmad, "A novel attack detection scheme for the industrial internet of things using a lightweight random neural network," IEEE Access, vol. 8, pp. 89337-89350, 2020.

[8] L.A. Tawalbeh, F. Muheidat, M. Tawalbeh, and M.Quwaider, "IoT Privacy and security: Challenges and solutions," Appl. Sci., vol. 10(12), pp. 4102, June 2020.

[9] M. Osman, J. He, F.M.M. Mokbal, and N. Zhu, "Artificial neural network model for decreased rank attack detection in RPL based on IoT networks," Int. J. Netw. Secur, vol. 23(3), pp. 496-503, April 2021.

[10] S.K. Sarma, "Optimally configured deep convolutional neural network for attack detection in internet of things: impact of algorithm of the innovative gunner," Wirel. Pers. Commun., vol. 118(1), pp. 239-260, January 2021.

[11] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," Simul. Model. Pract. Theory., vol. 101, p. 102031, March 2020.

[12] J. Canedo and A. Skjellum, "Using machine learning to secure IoT systems," 14th annual conference on privacy, security and trust (PST) IEEE. , pp. 219-222, December 2016.

[13] K. Filus, J. Domańska, and E. Gelenbe, "Random neural network for lightweight attack detection in the iot. In Modelling, Analysis, and Simulation of Computer and Telecommunication Systems", 28th International Symposium, MASCOTS 2020, Nice, France, pp. 79-91, November 2020.

[14] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: application areas, security threats, and solution architectures," IEEE Access, vol. 7, pp. 82721-82743, June 2019.

[15] S. Manimurugan, S. Al-Mutairi, M.M. Aborokbah, N. Chilamkurti, S. Ganesan, and R. Patan, "Effective attack detection in internet of medical things smart environment using a deep belief neural network" IEEE Access, vol. 8, pp.77396-77404, April 2020.

[16] A.K. Sahu, S. Sharma, M. Tanveer, and R. Raja, "Internet of Things attack detection using hybrid Deep Learning Model." Comput. Commun., vol. 176, pp. 146-154, June 2021.

[17] O. Yousuf, and R.N. Mir, "DDoS attack detection in Internet of Things using recurrent neural network," Comput. Electr. Eng., vol. 101, pp. 108034, May 2022.

[18] F. Hussain, S.G. Abbas, M. Husnain, U.U. Fayyaz, F. Shahzad, and G.A. Shah, "IoT DoS and DDoS attack detection using ResNet," IEEE 23rd International Multitopic Conference (INMIC), pp. 1-6, November 2020.

[19] H. Lin, and N.W. Bergmann, "IoT privacy and security challenges for smart home environments". Information, vol. 7(3), pp. 44, July 2016.

[20] S. Guergov, and N. Radwan, “Blockchain Convergence: Analysis of Issues Affecting IoT, AI and Blockchain” International Journal of Computations, Information and Manufacturing, vol. 1(1), 2021.

[21] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: a review," J. Big Data, vol. 6(1), pp. 1-21, December 2019.

[22] H. Mliki, A.H. Kaceam, and L. Chaari, "A comprehensive survey on intrusion detection based machine learning for IOT networks," EAI Endorsed Transactions on Security and Safety, vol. 8(29), pp. e3-e3, 2021.

[23] P. Williams, I.K. Dutta, H. Daoud, and M. Bayoumi, "A survey on security in internet of things with a focus on the impact of emerging technologies," IoT, vol. 19, pp. 100564, July 2022.

[24] M. Husamuddin and M. Qayyum, "Internet of Things: A study on security and privacy threats," 2nd International Conference on Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, pp. 93-97, April 2017.

[25] N. Narisetty, G.R. Kancherla, B. Bobba, and K. Swathi, “Investigative Study of the Effect of Various Activation Functions with Stacked Autoencoder for Dimension Reduction of NIDS using SVM,” Int J Adv Comput Sci Appl, vol. 12(5), 2021.

[26] N. Nirmalajyothi K. G. Rao, B. Bobba, K. Swathi, “Performance Analysis of Different Activation and Loss Functions of Stacked Autoencoder for Dimension Reduction for NIDS on Cloud Environment,” International Journal of Engineering Trends and Technology, vol. 69(4), pp. 169-176.

[27] K. Saurabh, A. Singh, U. Singh, O.P. Vyas, and R. Khondoker, “GANIBOT: A Network Flow Based Semi Supervised Generative Adversarial Networks Model for IoT Botnets Detection,” IEEE International Conference on Omni-layer Intelligent Systems (COINS), pp. 1-5, August 2022.

[28] https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT

[29] I. Ali, A.I.A. Ahmed, A. Almogren, M.A. Raza, S.A. Shah, A. Khan, and A. Gani, "Systematic literature review on IoT-based botnet attack," IEEE Access, vol. 8, pp. 212220-212232, November 2020.

[30] N. Nirmalajyothi, K.G. Rao, B.B. Rao, and K. Swathi, Performance of Various SVM Kernels for Intrusion Detection of Cloud Environment. International Journal of Emerging Trends in Engineering Research, vol. 8(10), 2020.