CORE
Provided by CORE

Metadata, citation and similar papers at core.ac.uk

Wright State University

CORE Scholar

Browse all Theses and Dissertations

Theses and Dissertations

2017

Deep Learning Approach for Intrusion Detection System (IDS) in the Internet of Things (IoT) Network using Gated Recurrent Neural Networks (GRU)

Manoj Kumar Putchala
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation

Putchala, Manoj Kumar, "Deep Learning Approach for Intrusion Detection System (IDS) in the Internet of Things (IoT) Network using Gated Recurrent Neural Networks (GRU)" (2017). *Browse all Theses and Dissertations*. 1848.
https://corescholar.libraries.wright.edu/etd_all/1848

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

DEEP LEARNING APPROACH FOR INTRUSION DETECTION SYSTEM (IDS) IN THE INTERNET OF THINGS (IOT) NETWORK USING GATED RECURRENT NEURAL NETWORKS (GRU)

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

By

MANOJ KUMAR PUTCHALA
B.E., Osmania University, 2011

2017
Wright State University

WRIGHT STATE UNIVERSITY

GRADUATE SCHOOL

July 27, 2017

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Manoj Kumar Putchala ENTITLED Deep Learning Approach for Intrusion Detection System (IDS) in the Internet of Things (IoT) network using Gated Recurrent Neural Networks (GRU) BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

Michelle A. Cheatham, Ph.D.
Thesis Director

Mateen M. Rizki, Ph.D.
Chair, Department of Computer Science
and Engineering

Committee on
Final Examination

Michelle A. Cheatham, Ph.D.

Adam R. Bryant, Ph.D.

Mateen M. Rizki, Ph.D.

Robert E. W. Fyffe, Ph.D.
Vice President for Research and
Dean of the Graduate School

ABSTRACT

Putchala, Manoj Kumar. M.S., Department of Computer Science and Engineering, Wright State University, 2017. Deep Learning Approach for Intrusion Detection System (IDS) in the Internet of Things (IoT) network using Gated Recurrent Neural Networks (GRU).

The Internet of Things (IoT) is a complex paradigm where billions of devices are connected to a network. These connected devices form an intelligent system of systems that share the data without human-to-computer or human-to-human interaction. These systems extract meaningful data that can transform human lives, businesses, and the world in significant ways. However, the reality of IoT is prone to countless cyber-attacks in the extremely hostile environment like the internet. The recent hack of 2014 Jeep Cherokee, iStan pacemaker, and a German steel plant are a few notable security breaches.

To secure an IoT system, the traditional high-end security solutions are not suitable, as IoT devices are of low storage capacity and less processing power. Moreover, the IoT devices are connected for longer time periods without human intervention. This raises a need to develop smart security solutions which are light-weight, distributed and have a high longevity of service. Rather than per-device security for numerous IoT devices, it is more feasible to implement security solutions for network data. The artificial intelligence theories like Machine Learning and Deep Learning have already proven their significance when dealing with heterogeneous data of various sizes. To substantiate this, in this research, we have applied concepts of Deep Learning and Transmission Control Protocol/Internet Protocol (TCP/IP) to build a light-weight distributed security solution with high durability for IoT network security.

First, we have examined the ways of improving IoT architecture and proposed a light-weight and multi-layered design for an IoT network. Second, we have analyzed the existing applications of Machine Learning and Deep Learning to the IoT and Cyber-Security. Third, we have evaluated deep learning’s Gated Recurrent Neural Networks (LSTM and GRU) on the DARPA/KDD Cup '99 intrusion detection data set for each layer in the designed architecture. Finally, from the evaluated metrics, we have proposed the best neural network design suitable for the IoT Intrusion Detection System. With an accuracy of 98.91% and False Alarm Rate of 0.76 %, this unique research outperformed the performance results of existing methods over the KDD Cup ’99 dataset. For this first time in the IoT research, the concepts of Gated Recurrent Neural Networks are applied for the IoT security.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

With blessings from my mother and father and along with immense support from my uncle Mr. Malleswara Rao, other family members and friends made me to reach to this point of life. Thanking them is the least I can do but I can glorify their love by trying to deliver my best at every stage of the life. I would like to thank Wright State University for providing me an opportunity to perform this research in my Master's program. I can confidently say that this research has enhanced my career goals.

I am also grateful to Dr. Michelle Cheatham for advising my thesis despite the pressure of timelines. She has guided me in the right path of research when the progress was stagnant. Without her, I would not have delivered the successful performance in my thesis. I am also thankful to Dr. Adam Bryant who has motivated me to aim high which prompted me to choose this unique thesis research. And, lastly, Dr. Mateen Riziki has been an excellent guide throughout my time at Wright State University.

1. Introduction

1.1 Overview

The world is currently witnessing the rapid product launches and high expectations from emerging Internet of Things (IoT) technology. It is growing at an accelerating pace connecting billions of devices in our daily life. As per the Gartner event analysis, there will be around 25 billion connected things by the year 2020 [1]. These connected devices enhance day to day activities and build smart solutions. But, the massive opportunities and utilities delivered by IoT technologies are shadowed by privacy trade-offs and grievous security concerns. One must consider the numerous connected devices, complexities, competing trends and diversities that must be managed while developing solutions for IoT. The current security protocols are only applicable for high powered computers for short-lived sessions. It is not viable to use the same protection technique for long-running sessions. For these reasons, IoT devices became attractive targets for the hackers making our lives endangered with unexpected threats.

One practical approach for dealing with these complexities of the IoT could be the use of the concepts of "lightweight" and "adoption" to develop robust security solutions. "Adaptive Lightweight" solutions have proven their worth many times in dealing with inconsistencies in very large distributed systems. It is almost impossible to design a security solution for each IoT device in a network because there are so many. However, securing the data that is transmitted between the devices in an IoT network would be a practical approach. With the help of artificial intelligence, wide ranges of sizes and types of data can be analyzed to develop adaptive solutions for the IoT system.

Machine Learning and Data Analytics techniques are already employed to improve customer service and network efficiency by analyzing the huge amounts of IoT data. Machine

Learning concepts such as pattern recognition, anomaly detection, and behavioral analytics can be used in IoT networks for detecting any potentially harmful behavior and blocking abnormal activities.

In this thesis, we came up with a unique research that involves a Multi-Layer architecture for an IoT system. We then applied deep learning algorithms to the IoT network for monitoring the network data to classify activity either as “normal” or “malware attack” for each layer in the architecture. We have used the KDD 99’Cup intrusion detection dataset [29] which is a benchmark data set used in most of the machine learning research on network data security.

1.2 Motivation:

Deep Learning ‘mimics the brain functionality’ with the help of robust neural network algorithms. The wide range of deep learning applications includes image recognition, computer vision, speech recognition, pattern recognition and behavior recognition. In the world of IoT, the datasets are high-dimensional, temporal and multi-modal. Deep Learning algorithms with robust computation power are more suitable for complex IoT datasets compared to legacy machine learning techniques. The application of deep learning to the IoT domain, particularly in IoT security is still in the initial stages of research and has a great potential to find insights from the IoT data. With smart use of deep learning algorithms, we believe that IoT solutions can be optimized. For example, recurrent neural networks in deep learning have the capability to learn from previous time-steps of the input data. The data at each time-step is processed and stored and given as input to the next time-step. The algorithm at the next time step utilizes the previous information stored to process the information. Though the neural network structures are complex, the hyperparameters can be tuned to obtain light-weight functionality for IoT solutions. This hypothesis motivated us to apply deep learning concepts to IoT network security.

1.3 Problem Statement

The goal of this thesis is to analyze and answer the following research questions:

- What are the security and privacy issues relevant to the IoT environment?
- Does GRU better than the other machine learning approaches for Intrusion Detection on the IoT?
- Does a separate GRU based IDS for each network layer perform better than the all layer GRU?

1.4 Purpose, Scope, and Contribution

The purpose of this thesis is to analyze the applications of deep learning to the Internet of Things (IoT) network security by evaluating recurrent neural network algorithms on the intrusion detection dataset. We believe that this research has an immense potential to open the doors for Deep Learning applications to both the cyber security domain and the IoT domain. The importance of security in today's connected world requires analyzing the humongous amount of heterogeneous data, and this cannot be possible without the help of artificial intelligence. This research can be extended by applying the algorithms on GPU environment on real-time IoT data.

Though there are various deep learning algorithms such as deep neural networks, auto encoders, convolutional neural networks and recurrent neural networks, the research problem requires an algorithm that can learn from historical data. Therefore, we have selected the family of recurrent neural networks for the research. Considering the need of building smart and light-weight solutions for the IoT network, we have performed the experiments with only the Gated-Recurrent-Unit (GRU) algorithm while the vanilla RNN and LSTM are ignored. However, we have evaluated various versions of GRU such as bi-directional GRU and multi-layer GRU to obtain optimized results on the dataset. Due to the unavailability of intrusion detection data for an IoT network, we have considered the KDD 99' Cup / DARPA dataset, which is used as a benchmark by most other IDS-Machine Learning researchers. However, we have modified the data by dividing it into various layers such that the same procedure can be applied in an IoT network.

As this is an inter-disciplinary research project which involved cybersecurity, artificial intelligence and computer networks, a lot of time has been spent in understanding the depth of the concepts in each field. We started with understanding the attack types in an intrusion detection dataset. We followed up with learning the architecture of TCP/IP and analyzed the possible malware attacks at each layer. We then realized that the intrusion detection data-set must be classified with machine-learning algorithms. We also learned the IoT architecture and started evaluating machine learning algorithms satisfying the IoT characteristics. However, we recognized that the application of deep learning algorithms is the most suitable approach for the research problem defined. We performed the experiments using a robust deep learning tool called Google's Tensorflow. We have also applied the higher versions of recurrent neural networks. The performance results are compared for each designed IDS layer with All-layers IDS. This overall interdisciplinary practical approach made this research unique.

1.5 Research Methodology

We started with a rigorous literature review on the current security and privacy issues in the IoT. We then framed the characteristics which are apt for building IoT solutions. In the next step, we proposed a multilayer approach that satisfies the defined IoT characteristics. As a proof of concept, we have selected the IoT network data security issue to design a smart solution. We have applied deep recurrent neural network algorithms on the network data to classify each sample as "normal" or "intrusion". Due to the IoT data unavailability, we have used the IDS benchmark data-set as it provides us an opportunity to compare the results of this thesis with existing results on that dataset. We have evaluated the performance of algorithms using accuracy, precision, recall, f1 score, false alarm rate, the area under ROC curve and the confusion matrix. We then compared the results at each layer of the designed IoT architecture with the all-layers IDS and existing literature. The overview of the methodology of this research is summarized as below in Figure 1.

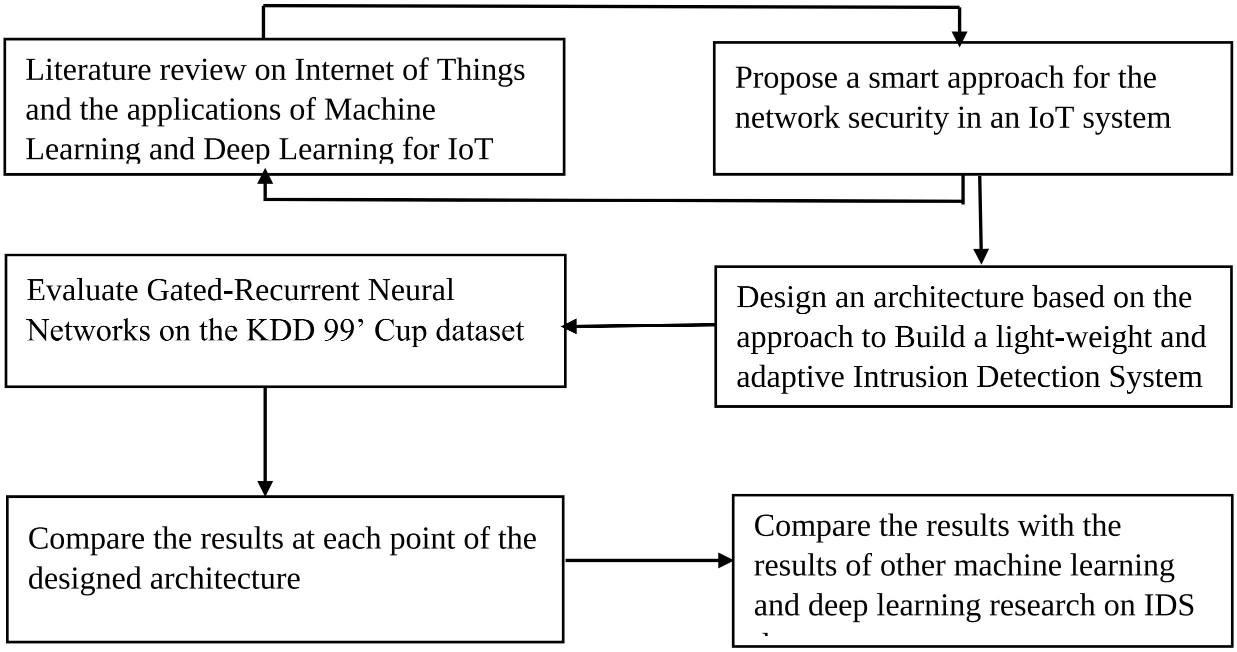

Literature review on Internet of Things and the applications of Machine Learning and Deep Learning for IoT
Propose a smart approach for the network security in an IoT system
Evaluate Gated-Recurrent Neural Networks on the KDD 99' Cup dataset
Design an architecture based on the approach to Build a light-weight and adaptive Intrusion Detection System
Compare the results at each point of the designed architecture
Compare the results with the results of other machine learning and deep learning research on IDS

Figure 1: Research Methodology

2. Background

This chapter provides information on the background involved in developing this research. It introduces the concepts and technologies of Internet of Things (IoT) followed by its security and privacy concerns. Further, this chapter addresses the characteristics to be considered while developing IoT security solutions. This chapter continues to elucidate the concepts of network security and intrusion detection systems. Later, explanation of the network architecture is provided for the Long-Short-Term-Memory (LSTM) and Gated-Recurrent-Unit (GRU) recurrent neural networks. The final sections of the chapter explain the importance of Machine Learning and Deep Learning for IoT security by quoting their relevant applications. The main intention of this chapter is to provide a complete overview of the concepts and algorithms for the reader with minimal knowledge in this field of research.

2.1 What is the Internet of Things?

The Internet of Things (IoT) incorporates everything in the world, from the body sensor to modern cloud computing. It ubiquitously connects machines, networks, and humans, thus creating a complex distributed system. It advances human life by achieving robust machine-to-machine communication and machine-to-human communication. In this context, the IoT introduced smart grids, smart homes, smart cities and the Industrial Internet of Things (IIoT). Due to its wide range of applications and diverse technologies, the IoT has created many research opportunities in recent years.

2.2 Technologies in IoT

The reality of the IoT is possible by integrating various enabling technologies. Major contributors to the IoT are sensors, radio-frequency identification (RFID), nanotechnologies and smart technologies as shown in Figure 2

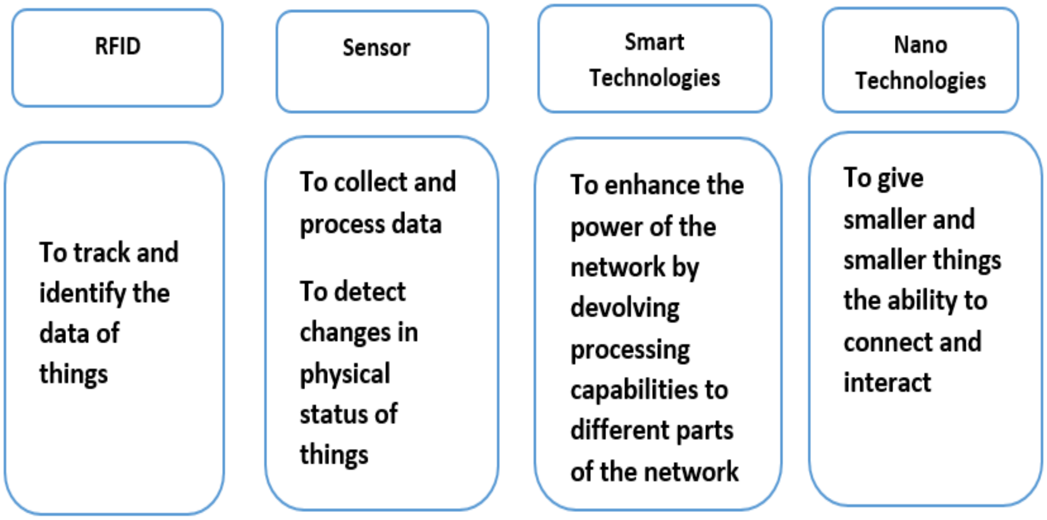

RFID
Sensor
Smart Technologies
Nano Technologies
To track and identify the data of things
To collect and process data
To detect changes in physical status of things
To enhance the power of the network by devolving processing capabilities to different parts of the network
To give smaller and smaller things the ability to connect and interact

Figure 2: Underlying Technologies of Internet of Things(IoT)

RFID: Radio-Frequency Identification (RFID) devices are microchips which are wireless for automated identification and tagging objects. These devices can identify the object without line-of-sight wirelessly, with the use of a reading device called a reader and tags to detect the channel and sense the collections. RFID technology is used in various current day applications like credit cards, automobile ignition keys and so on. The application of RFID technologies to the Internet of Things (IoT) system is crucial to exploit the moving nodes and to build intelligent systems.

Sensor: The Internet of Things (IoT) is impossible without the use of sensors in the system. The communication between the IoT ecosystem is achieved with data flow between the devices where the data is collected and received. Sensors act as a gateway to collect the data and to detect the physical status of the things.

Smart Technologies: Smart technology devices like smart fridge, smart phone, and other wearable technologies make the Internet of Things(IoT) dream possible with robust performance in the network. The smart technologies adapt smart solutions while accessing the resources in the IoT system and enhance the processing capabilities of the network.

Nano Technologies: Complex IoT systems make use of nanotechnologies, which have a potential impact to design smart solutions. For example, nanosensors can be used in city locations to monitor the spread of diseases.

2.3 Privacy and Security Issues in IoT:

Even though the Internet of Things (IoT) is a boon to the society, it also creates serious security and privacy concerns. With the day-to-day interactions with real-time applications and with most of the IoT devices left unattended without any monitoring, the IoT system raises numerous privacy and security concerns. The range of vulnerabilities that the IoT raises is vast in terms of infrastructures, network, device, and interface. Based on the exposure of the network, the issues in privacy and security can be categorized as shown in Figure 3.

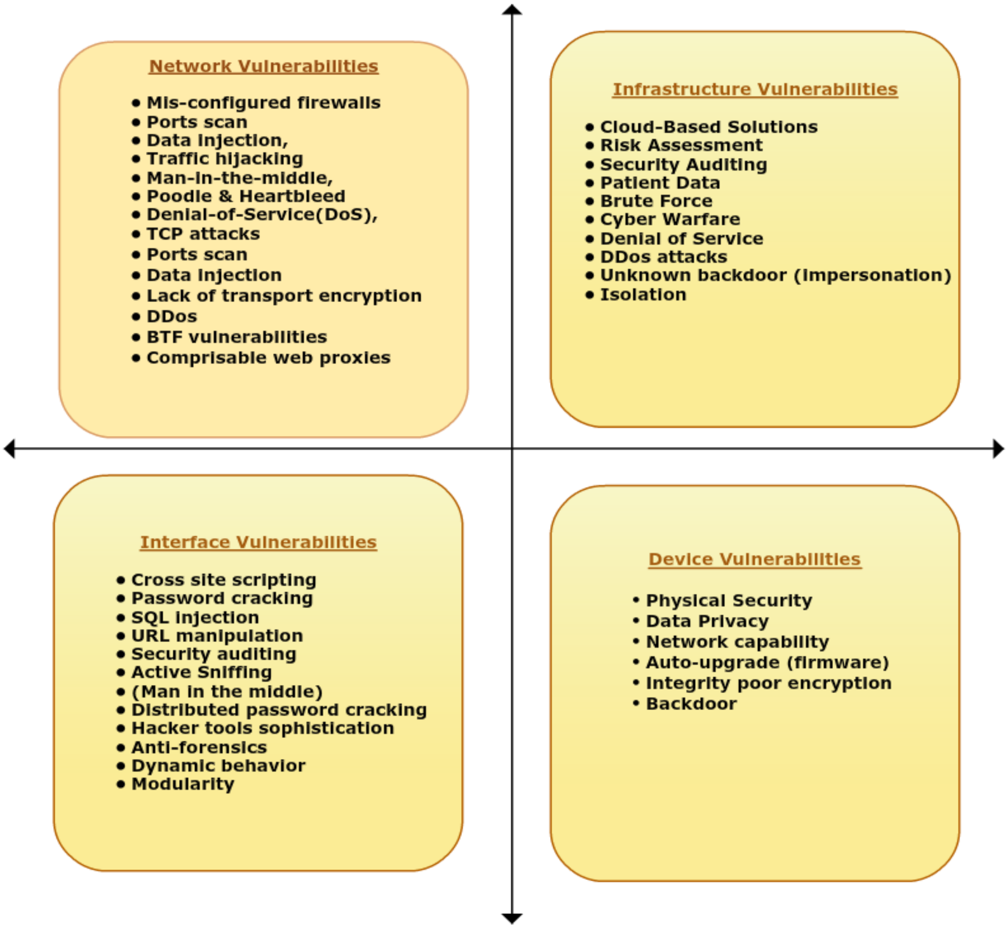


Figure 3: Security Threats in IoT with respect to architecture

2.4 Importance of network security in IoT:

The size and count of the devices in the IoT network makes it difficult to implement per-device security. Network-based security can be implemented to act as a protective shield by monitoring the data throughout the network. Moreover, the network based solutions can be easily applied to other IoT networks with minor required changes, unlike device security. The devices in the IoT network should be registered to allow access to the network to be protected from intruders. All the incoming and outgoing traffic for each device needs to be monitored and a template should be created for the normal behavior of the network traffic. Any network data which does not fall in the established behavior is identified as a threat and alarms are signaled to the owners of the devices.

2.5 Intrusion Detection System (IDS)

Network security can be achieved with the help of a software application called an Intrusion Detection System (IDS), which monitors the malicious activities of the system or the networks. IDS can be classified into different types. Based on their responsive nature, IDS is categorized into Active IDS and Passive IDS. An Active IDS is designed to block the malware attacks automatically, without any human intervention, whereas a passive IDS only monitors the network traffic and alerts the users. Another categorization of IDS is Signature-Based IDS and Anomaly-based IDS. In the signature based approach, the IDS access a database of known signatures and vulnerabilities. Each intrusion attack contains the complete details of the attack called a signature and that is used to detect and prevent future attacks. The main drawback of this method is that the database needs to be updated frequently, whereas the anomaly-based IDS (behavior-based) learns from the baseline patterns to detect new intrusion attacks. Any deviation from the existing baseline patterns are identified as attacks and alarms are triggered. Another division of the IDS is based on the place where it is mounted. When an IDS is placed on the network segment it is said to be a Network Intrusion Detection System, whereas, when an IDS is deployed in workstations, they are said to be Host-based Intrusion Detection Systems. Host-based Intrusion Detection Systems have a significant number of drawbacks and may not be suitable for research purposes.

2.6 Machine Learning and Deep Learning

Machine Learning is a branch of artificial intelligence that deals with the analysis and construction of systems from the knowledge gained from the data [3]. The wide range of applications of machine learning includes regression, classification, prediction and so on. It is categorized mainly into three types based on the use of labeled data– a) Supervised learning, b) Unsupervised learning, and c) Semi-supervised learning. The commonly used algorithms in machine learning include linear regression, Navie-Bayes classifier, logistic regression, support vector machines, artificial neural networks and so on.

Deep Learning is a complex version of machine learning with multiple levels of abstraction of data at multiple processing layers [4]. Deep Learning can learn the intricate structures in the dataset through back-propagation and indicates how machine changes the internal parameters at each layer. The frequently used deep learning algorithms include deep belief networks, auto-encoders, convolutional neural networks and recurrent neural networks.

2.7 Why is Deep Learning better than Machine Learning for the IoT?

Deep learning, which is also known as hierarchical learning or deep structural learning, is a broader version of machine learning in terms of complexity in the structure and learning data representations. The key difference between machine learning and deep learning is the change in the performance as the scale of the data increases. Deep Learning algorithms require a larger amount of data to find the patterns in the network where machine learning requires the less data. Artificial Neural Networks (ANN) that contain one or more hidden layers will make the structure deep and the data is processed at each layer, thus, making the learning task deeper. The commonly used deep learning architectures include deep belief networks (DBN), deep neural networks

(DNN) and recurrent neural networks (RNN), which are applied to research fields such as natural language processing, speech recognition, computer vision, audio recognition, machine translation and social network filtering. Deep Learning can also be applied to IoT data research where the data is heterogeneous and multi-modal. Traditional machine learning algorithms fail to deliver long term results for IoT devices which are usually connected for longer time-periods. Recurrent neural networks (RNN) in deep learning have the capability to learn from the previous time-steps and can be used with less human intervention. In RNN, the output of each node in the hidden layer is given as input to the same node at each time-step. The useful information is stored in the memory and can be used for learning purposes in future time steps. The difference between an RNN and Feed-forward neural network(FNN) can be seen in Figure 4:

Figure 4: Hidden Units in Recurrent Neural Network vs Feed Forward Neural Network

2.8 Long-Short-Term Memory RNN (LSTM):

Recurrent Neural Networks (RNN), when trained in real-time learn from previous time-steps by backpropagation through time (BPTT). A deep neural network is unfolded in time and constructs an FNN for every time-step. Then, the gradient rule updates the weights and biases for each hidden layer, thus, minimizing the loss between the expected and actual outputs. However, standard RNNs cannot perform better when the time-steps are more than 5-10. The prolonged back-propagation leads to vanishing or blow up of error signals, leading to oscillating weights, which makes the network performance poor. To overcome this vanishing gradient problem, researchers came up with the Long-Short-Term-Memory (LSTM) network which bridges the minimal time gaps. LSTM makes use of a gating mechanism to handle long-term dependencies. The LSTM structure can be seen in Figure 5:

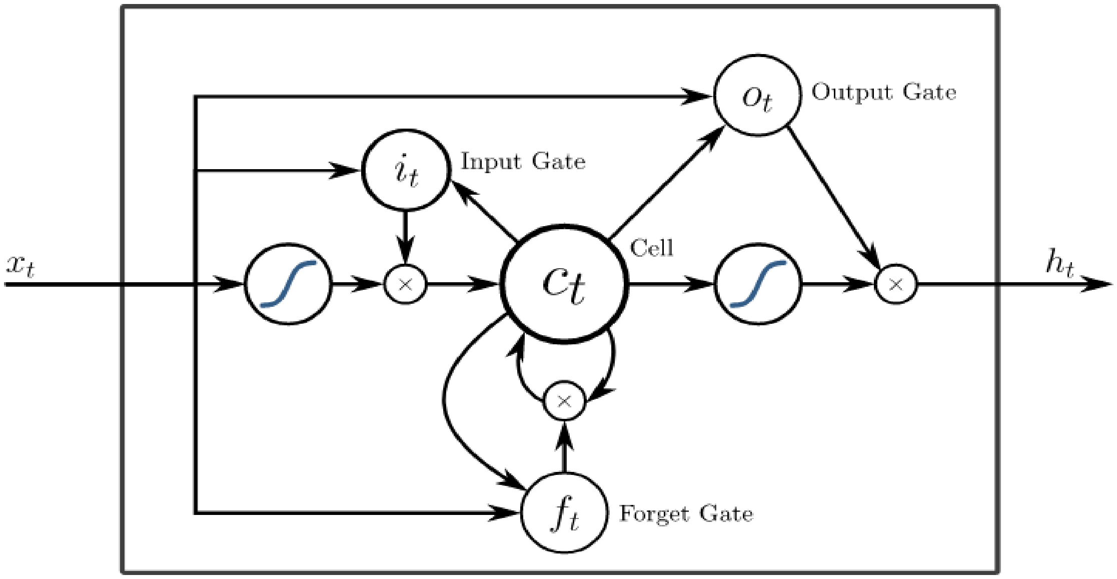


Figure 5: Long Short Term Memory Cell

LSTM has a cell state which is passed to every-time step. A gating mechanism is used to optimize the information that is passing through. It contains a sigmoid function layer which outputs between one and zero. A value of one means “pass all the information through”, whereas the value of zero means “do not pass any information through”. The "forget gate" decides the information that needs to be let through between the current input and previous cell state output using the sigmoid function. The "input gate" decides what information is required to store in the cell state. This gate contains two functions - "sigmoid" to decide what values need to be updated, and the “tanh” function to create a new vector of values that are to be added to the cell state. The “output gate” decides on what information from the cell state is required to output with the help of a sigmoid function. The output information is passed through the "tanh" function before passing through the "sigmoid", to make sure that the values are between -1 and +1.

2. 9 Gated Recurrent Unit (GRU):

A Gated Recurrent Unit (GRU) is a lighter version of an LSTM where the complexity in the structure is reduced by decreasing the gates in the architecture. The GRU merges both the "forget gate" and "input gate" in an LSTM to an "update gate" and combines the hidden state and cell state, resulting in a simpler architecture of the network as shown below Figure 6:

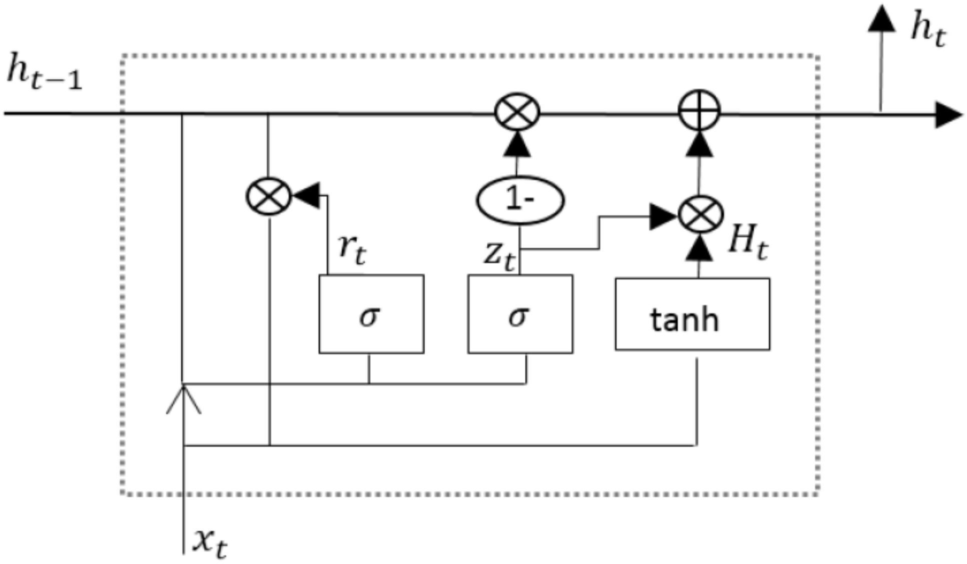


Figure 6: Gated Recurrent Unit Cell

2.10 Multi-Layer GRU RNN:

Deep recurrent neural networks can have various architectures which influence greatly the performance of the algorithm. One can add many layers of RNN (plain RNN, LSTM or GRU) cells and stack the network into a deep structure called a Multi-Layer RNN as shown in Figure 7. This technique has a wide range of applications in speech recognition systems and weather forecast systems with high dimensional data. When GRU cells are used in each hidden layer of a recurrent neural network it is said to be a Multi-Layer GRU. In the multilayer structure, the input of the network is passed through multiple GRU layers apart from back propagation through the time. It has been proven that multi-layered RNNs learn from the different time lengths of input sequences [4]. Another key important feature of multi-layered RNNs is that they share the hyperparameters, weights, and biases across the layers, thus achieving optimized performance.

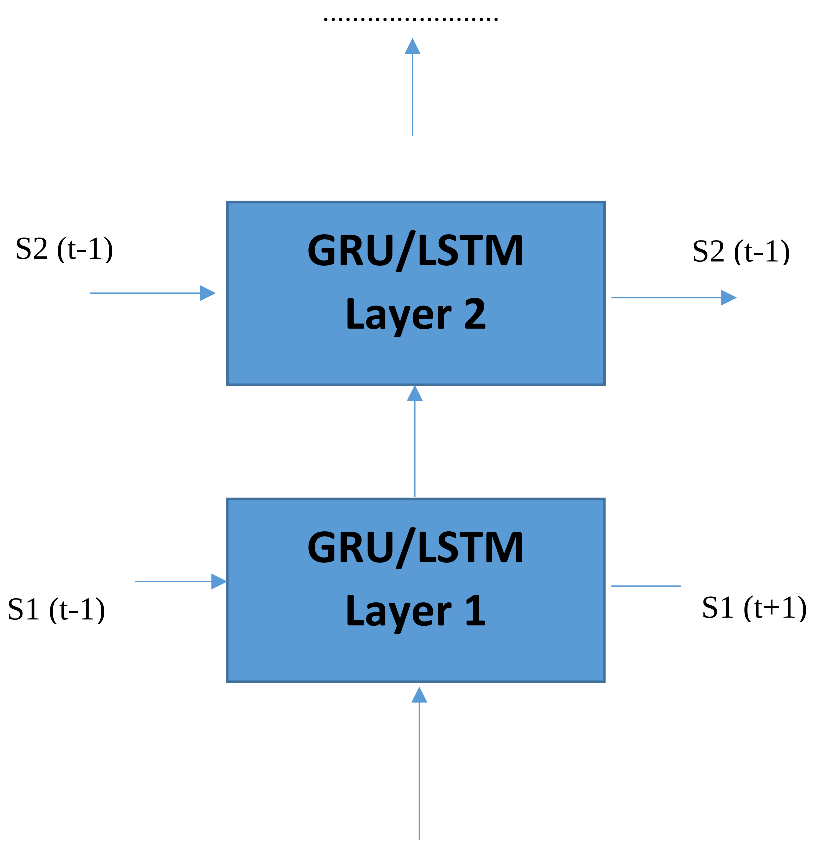


Figure 7: A Multi-Layered LSTM/GRU recurrent neural network

2.11 Hyper-Parameters:

The hyperparameters used in the design of the recurrent neural network have a great impact on the performance of the network [5]. Although there are many hyper-parameters involved in the design of a recurrent neural network, the parameters having the largest impact on the performance of the network are learning rate, number of hidden layers, number of units/cells in the hidden layer and the number of time-steps.

Learning Rate: It is a measure of the rate at which the network optimizes the minimization of the loss function in a neural network. Mathematically, if the loss function is L (X; W, b), then the goal of the network is to minimize the loss (cost) function L. The weights are constantly updated to achieve the best possible output reducing the loss value. The learning rate determines how fast the parameters are updated. One must vary the learning rate during the training of the neural network to obtain the best results.

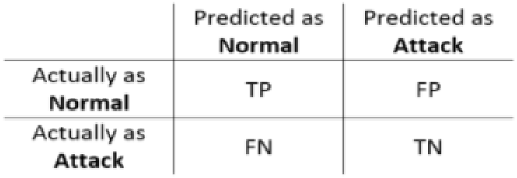
Time-Steps: Selecting the number of time-steps also plays a crucial role in the performance of the system. The information required to find the correct patterns depends on the number of time-steps that are required to back propagate. Tuning the number of time-steps improves the output of the network. When more time-steps are selected, the network takes longer to time to train and vice-versa.

Hidden Units: The number of cells in a hidden layer determines the amount of computation performed on the input data [6]. The more hidden units in the network, the longer it takes to train. The neural network should be trained for a various numbers of hidden units to verify the performance of the system.

Hidden Layers: The stacking of GRU layers as discussed in the above multilayer GRU section, has a great impact on higher dimensional datasets. However, most deep neural networks obtain optimized performance with a single hidden layer [6]. One must decide on the number of hidden layers to be used with respect to their data-set size and the dimensions.

2.12 Evaluation Metrics

To evaluate the performance of the classification model the following metrics are used in machine learning research. In general, the confusion matrix visualizes the performance of the algorithm in a tabular form as shown in the figure below:



	Predicted as **Normal**	Predicted as **Attack**
Actually as **Normal**	TP	FP
Actually as **Attack**	FN	TN

Where,

- True Positive (TP) is the total number of samples predicted as "normal" while they were "normal".
- False Negative (FN) is the total number of samples predicted "normal" while they were "attack".
- False Positive (FP) is the total number of samples predicted "attack" while they were "normal".
- True Negative (TN) is the total number of samples predicted "attack" while they were "attack".

All other important metrics such as Precision, Accuracy, Recall, False Alarm Rate (FAR) and Area under ROC curve (AUC) can be calculated using these 4 measures taken from the confusion matrix as shown below:

$$recall = \frac{TP}{TP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

$$accuracy = \frac{TP + TN}{(TP + FP) + (FN + TN)}$$

$$FAR = \frac{FP}{FP + TN}$$

2.13 Random Forest Classifier (RF)

A Random Forest Classifier is an ensembled machine learning technique for supervised learning tasks. This algorithm was initially suggested by Breiman where he described the advantages of Random Forest as below [21]:

- Ability to handle numerous input variables without a necessity for variable deletion.
- Can run on huge data bases efficiently
- Provides estimates of important variables for the classification
- Robust to noise and outliers when compared to single classifiers
- Lightweight when compared to other boosting methods

We have made use of the ability of the random classifier method to rank the importance of the features set to the target variables. We have selected those variables based on the maximum importance levels. Those features with low values of the importance will add less information to the learning model and are ignored based on the threshold values of the importance.

3. Literature Review

3.1 Machine Learning for Cyber-Security

With the rapid advancements in cyber-attacks and availability of huge amounts of malicious data in cyber infrastructures, machine learning and data mining and other interdisciplinary capabilities are frequently used to address the challenges of cyber security. Machine learning can be applied in signature detection, anomaly detection, scan detection, network traffic profiling and privacy-preserving data mining.

Machine Learning for Signature/Misuse Detection: Signature Detection, also called misuse detection, is recognizing unique patterns of unauthorized behavior to detect and predict future similar attempts [2]. To handle uncertainties and improve robustness fuzzy-rule-based techniques are used. When new attacks occur, these fuzzy-rule systems generate human-like expertise in decision making rather than relying on humans to update [2].

Machine Learning for Anomaly Detection: Finding any event of the system falls outside of a predefined set of normal behaviors is the goal of anomaly detection. Clustering-based anomaly detection was done by Portnoy et al (2001), who found that the unlabeled data has more potential in detecting unknown attacks through a semi-automated or automated process, allowing cyber security experts to focus on the most likely attack data [2]. Zhang 2006, applied random forests to the DARPA MIT KDD Cup 1999 data set and evaluated the performance using ROC. The best detection rate was achieved by keeping the lower FP rate when compared to other unsupervised anomaly systems [2].

3.2 Machine Learning and Deep Learning for IoT:

Smart homes: Smart home is one of the most sensitive IoT environments as it involves users and daily use devices. The security and privacy must be at top-notch when it comes to IoT intelligent homes. Advanced deep learning techniques like Deep Belief Network-Artificial/Recurrent Neural Networks (DBN-ANN/RNN) can perform behavior analysis on living habits and provide continuous service with smart home automation [7].

Smart City: With smarter infrastructure and signal timing adjustments and dynamic re-routing, traffic flow can be improved. Despite all the hype, the smarty city faces potential cyber threats which can be influenced mainly by technology, governance, and socio-economic factors. The deployment of sensors, meters and the integration of real-time information from the citizens' results in a tremendous amount of data. The machine learning approach involves a huge amount of network traffic flows and detecting patterns in the huge traffic data. Network profiling contains the steps data capturing, filtration and generation of associated rules. The privacy of the data being the most crucial part as smart city network involves a large number of devices and citizens, machine learning uses privacy-preserving data mining(PPDM) algorithms to ensure security to the data [8].

Industrial IoT: The Industrial Internet of Things(IIoT) is the part of the Internet of Things (IoT) that focusses on devices and objects used in business settings. Devices can be used to sense and collect data from maritime fleets, to help reduce unplanned downtime or for manufacturing systems to provide better control processes. These IIoT systems generate large amounts of privacy-sensitive and security-critical data that needs to be protected in an efficient way. Luo X 2016, applied kernel machine learning algorithm in an Industrial IoT setting to predict large-scale web Quality of Service (QoS) [9].

Smart Grid: Smart grid encompasses the modernization of the power transmission and distribution grids that exist within the current power grid. This approach aims to improve the use of resources, distribute information and assets across organizational and geographical boundaries, integrate all critical information, and provide security from hazardous threats. A variety of attacks like eavesdropping, traffic analysis, replaying, spoofing, cracking and destruction can be performed by attackers on smart grid systems. Esmalifalak, M 2014, proposed a Support Vector Machine based method to detect the "stealth" attacks in the IoT smart grid [10].

Health-Care IoT: With a growing population, the IoT becomes crucial by being able to track every patient, piece of equipment and batch of drugs to improve efficiency and safety by extending the care beyond the hospital. In the same way, the IoT has caused new security issues in the healthcare industry which requires reformed measures to handle them. A few risks include possible harm to patient's health and safety, loss of infrastructure and physical attacks. An attacker that successfully hacks an IoT healthcare device can obtain access to a patient's sensitive information including his movements, health, and habits, making security and privacy concerns to the end users. H. Abie and I. Balasingham in their paper "Risk-Based Adaptive Security for Smart IoT in eHealth", described an adaptive security monitoring framework along with a case-study to support. The model consists of five important steps- 1) Identify the threat 2) Analyze the problem and predict the impact 3) Plan the corresponding actions 4) Track the focus on risk mitigation actions 5) Control the risk exposure [1]. At every point of this model, robust machine learning algorithms are used to provide sufficient security to the patient using the IoT Health care device [11].

3.3 Machine Learning applications on IDS:

Many researchers have already applied machine learning techniques to intrusion detection. Li, Y., & Guo, L. (2007), carried out a supervised active learning method based on Transductive Confidence Machines for K-Nearest Neighbors (TCM-KNN algorithm) for intrusion detection. Their research could achieve a false positive rate of 0.027 which can be improved further [13]. Another unsupervised method using clusters was proposed by Leung, K., & Leckie, C. (2005, January), with best Area Under ROC (AUC) of 0.973 [14]. Ashfaq, R. A. R., (2017), applied a fuzzy based semi supervised technique for intrusion detection datasets and achieved an accuracy of 84.12% and 68.62% on the KDD Cup ’99 test datasets. However, better-improved results on larger intrusion detection data sets are obtained using deep learning techniques that can be reviewed in section 3.4

3.4 Deep Learning applications on IDS:

Deep belief networks, auto-encoders, and restricted Boltzmann machines are widely used for the feature extraction for the intrusion detection data sets. Tao, X., (2016), used a supervised Fisher and Deep Auto-Encoder to extract the important features required for the KDD Cup '99 dataset [16]. Recurrent neural networks are highly used for classification and regression tasks, many the researchers have applied LSTM and GRU for IDS classification. Kim. J., (2017 February) has applied the Long-Short-Term-Memory algorithm along with Gradient Descent Optimization for an effective intrusion detection classifier with an accuracy of 97.54% and recall of 98.95% [17]. The same research team applied the Gated Recurrent Unit for the first time in the research on intrusion detection data sets with satisfactory results where the values of recall, false alarm rate, and accuracy are 97.06%, 10.01% and 98.65% [20]. Staudemeyer, R. C., (2013, October) evaluated the performance of LSTM networks on the KDD 99 ‘Cup IDS data set with satisfactory results [21]. The same team came up with improved results in which the cost of training the network and training accuracy were 22.13 and 93.82% [22]. All the existing research has considered random records with a small size, which is not apt for deep recurrent neural networks. We have tried to overcome this drawback by applying gated recurrent networks to the whole dataset.

4. Design and Architecture

4.1 Introduction

This chapter provides a brief information to the design approach and the datasets used in building an intelligent model for an IoT network. The key features required for IoT solutions are adaptive, lightweight, multi-layer, distributive and ability to learn from the past. We designed an innovative architecture for an IoT home network that would reduce the size of the datasets for the IDS classifier. We have selected the KDD Cup 1999 Intrusion Detection Dataset for the experiments and proposed an intelligent solution which satisfies the key requirements of the IoT solutions. We have performed the feature engineering using a Random Forest classifier and selected those features with high importance. We performed a rigorous data analysis and prepared the data in the required format before it was used as an input to the model.

4.2 Key features for IoT solutions

Internet of Things (IoT) security solutions are multi-faceted, where the data flow is protected with integrity, confidentiality, and authentication services; the system is secured against disruptions and intrusions. An IoT system deals with various heterogeneous devices and multi-modal data over time and hence, standard solutions may not be effective. There is a need to develop smart solutions which are applicable to various levels of data flow in the network. It raises an urge to invent scalable solutions to apply on devices with various memory sizes. We came up with three key features that are exactly required to handle the IoT systems as described below:

4.2.1 Light-Weight:

Unlike smartphones and computers, a significant number of IoT devices are not capable of processing anti-malware software due to the low-end operating system used in such devices. They do not have the infrastructure to perform advanced techniques to protect from sophisticated malware threats and at the same time, they lack sufficient memory space to store the ever-expanding list of malware databases. Lightweight security solutions make it easier for developers to roll out any security updates and to collect the performance of the devices which are used to take an action if new services or products are required to improve the performance. Thus, lightweight solutions make the system dynamic, scalable, performance optimized, interoperable and flexible.

4.2.2 Multi-Layered (Distributive):

The differences in the capabilities of IoT endpoint devices highlight the idea of a multi-layered distributive approach in the IoT architecture. With an open traditional architecture, the IoT system is easily prone to information, privacy and security leaks. The multi-layered architecture deals with devices and their data at various levels, making the system robust. In an IoT network, data is produced by various kinds of devices, processed and stored in different ways, and transmitted to various locations. A single layer model may not generate optimized performance across the IoT system, restricting the locality or scope of the components; whereas, a multi-level or multi-layered architecture is distributed across the system, allowing the processes to be executed at each level, from complex to trivial, based on the situation.

4.2.3 Longevity:

In terms of the Internet of Things, the longevity levels are different from typical handheld consumer devices. For example, in a smart city application, the life cycle of the device would be 10 to 20 years. Therefore, the IoT solutions designed for those devices should have the capability to learn continuously from past experiences without human assistance. The life span of IoT applications like smart TVs and refrigerators tend to have longer time-periods and must function in the absence of the manufacturer. Longevity is important to gather any demand for the IoT system devices as it improves the Return on Investment(ROI). It costs a lot for any agency to monitor the maintenance of the deployed devices in an IoT environment for a long time. There is a significant amount of research on developing IoT solutions which can work for more than 10 years with good longevity. Moreover, security solutions when processed for longer periods should have the capability to accommodate newer malware attacks over the period. Hence, there is an urge to improve the existing solutions in a smarter way.

4.3 Architecture:

Considering the above features for IoT security solutions, we have come up with a robust architecture to monitor intrusion detection activity in a sophisticated manner. Out of various security measures, we have selected network security as the use case to prove the defined features are apt for an IoT network. As we are going to deal with IoT network security, the research is performed on an intrusion detection data which contains information about “normal” and “malware” connection types logged in an IoT network. In a regular wireless system, the Intrusion Detection System (IDS) monitors the network data using either a “Signature-based approach” or an “Anomaly-based approach”. The IDS mounted at a point in the network obtains all the network data and classifies the data into “normal” or “attack”. Other than traditional approaches, Machine Learning (ML) algorithms are applied to a dataset and classification is performed through supervised learning. However, this legacy approach may not be suitable for smart IoT network systems due to their heterogeneity. As mentioned in section 4.2, the security solutions for intrusion detection should be light-weight, multi-layered and have a good amount of longevity. Hence, we developed a multi-layered architecture and applied light-weight machine learning algorithms which can work with better performance for longer periods of the time.

An IoT system contains various devices which are placed at different locations with long distances between them. The number of devices involved in IoT systems is higher when compared to a regular wireless or wired system. A single IDS system must have the memory capacity to process the network data among all the devices and must be responsive in a short amount of time. In this case, the performance will be poor in the IoT network due to the high number of devices and the large distance between the devices. Therefore, we have come up with an architecture where an IDS for the whole system can be replaced by four Intrusion Detection Systems based on the malware attacks that occur at each TCP/IP layer. Each TCP/IP layer has specific devices, for example, the transport layer contains switches, while the network layer contains routers. Each IDS placed at a TCP/IP layer monitors only the data obtained from the devices that belong to that layer. This way, the network load will be shared the system, so that it becomes light-weight and the response time improves. The architecture can be seen in the Figure 8 below:

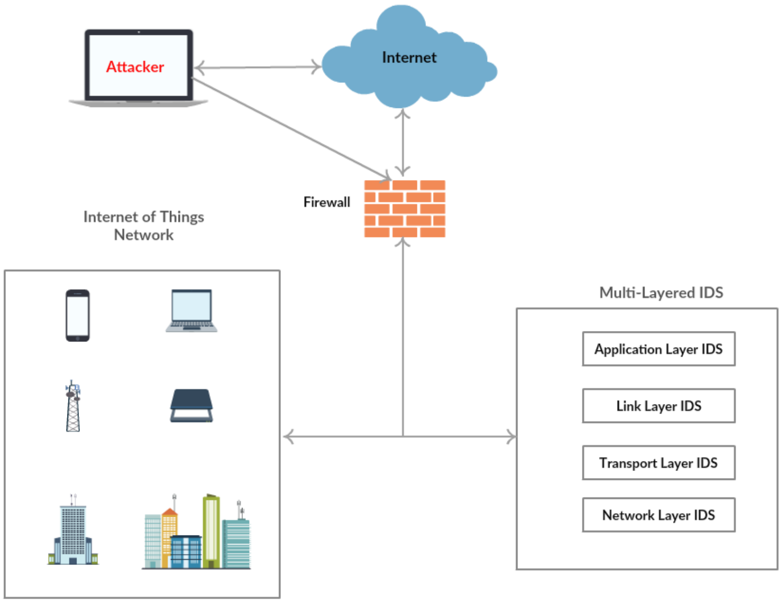


Figure 8: Multi-Layer architecture for IoT network

The next key features that need to be handled in developing an IoT security solutions are the "light-weight" and "Longevity" characteristics. We originally selected an anomaly detection approach where we planned to apply and evaluate various machine learning algorithms that occupy less memory. However, “Longevity” cannot be achieved when traditional machine learning algorithms are used because they do not have the capability to learn from past experiences. The entry of deep learning algorithms like recurrent neural network algorithms opened the path to develop solutions that can learn from past occurrences. Recurrent neural network algorithms such as Long-Short-Term-Memory (LSTM) and Gated-Recurrent-Unit (GRU) had already proven a great success in various domain applications. As discussed in Chapter 2, the GRU has less complexity in the architecture when compared to LSTM, thus, making it light-weight. We have performed the anomaly detection experiments using GRU and LSTM and evaluated the performances on the intrusion detection data set obtained at the various IDS layers. We have also applied improved versions of GRU algorithms like Dynamic GRU, Bi-directional GRU and obtained the optimized performance and results for the dataset. Thus, the GRU algorithm has the capability to satisfy both the key features of IoT solutions which are “light-weight” and “Longevity”. The results obtained at each layer are compared with results obtained for single layer IDS, and are explained in detail in Chapter 4.

4.4 IDS - Datasets:

Intrusion detection data for training machine learning algorithms are limited in the literature. The two mainly used datasets are the UNB ISCX 2012 datasets and KDD Cup'99 /DARPA datasets. The most literature on the application of machine learning algorithms on intrusion detection data set uses the DARPA KDD Cup '99 dataset and hence we selected this for the research. This way, the results obtained in this research can be compared to the results of the previous research.

The DARPA KDD Cup '99 datasets were generated by the Defense Advanced Research Projects Agency (DARPA ITO) on a simulated air force model. The training data was collected for seven weeks and the testing data were collected for two weeks. The whole dataset contains 39 network-based attack types and has more than 200 instances of background traffic compared to an air force base model. The complete network traffic is either classified as one of the attack types or "normal". The datasets can be found on the UCI website where repository links to the three different versions of data set exist. The three versions of the KDD 99'Cup IDS datasets are – full KDD data set, corrected KDD, 10% KDD. Among these three, 10% KDD data set is used in most literature and hence, we are using the same for this research. As discussed before, using the same dataset which was used before will provide a chance to compare this thesis results with the existing results. The 10% KDD dataset contains 24 attack types, which are mainly categorized into four classes – Probe, Denial of Service(DoS), User to Root (U2R) and Remote to Local (R2L). The training and testing samples are represented with 41 features and a label with either "normal" or "attack type". The features can be divided into three types: the first group describes the features that are used for providing information on the command that is used for connections, the second group of features describes the specifications of the commands, and the third group describes the features that convey information about the connections having the same destination with the same service. In most of the literature that uses the intrusion detection dataset, the researchers randomly selected a specific amount of records and used it for training and testing their models. However, in this thesis, we considered the whole dataset. As the GRU algorithm requires a time series dataset, we have neither randomized the sequence nor removed the duplicates, thus, making it apt for GRU classification analysis.

4.5 Data Flow:

The end to the flow diagram of the deep machine learning model is represented in Figure 9.

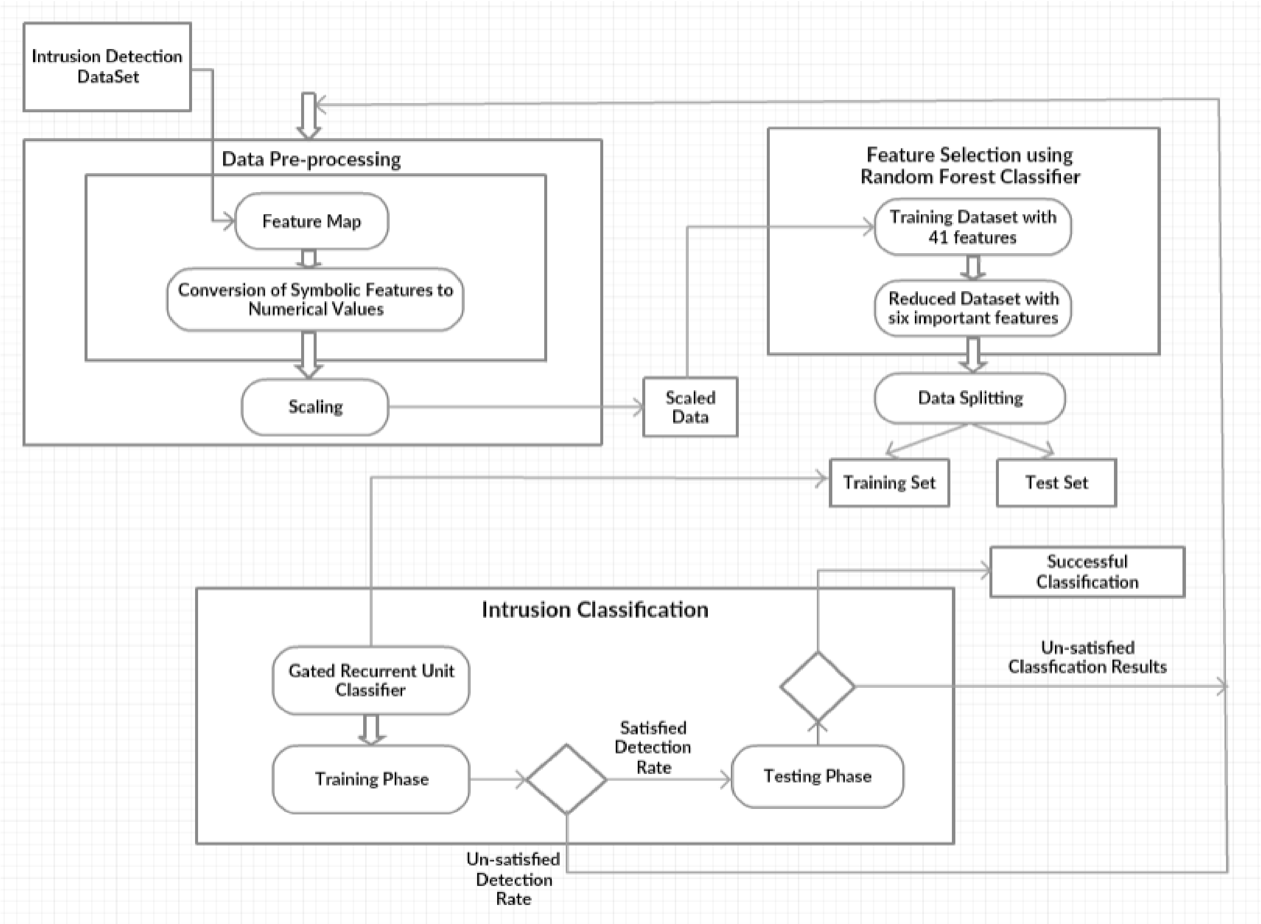

Intrusion Detection DataSet
Data Pre-processing
Feature Map
Conversion of Symbolic Features to Numerical Values
Scaling
Scaled Data
Feature Selection using Random Forest Classifier
Training Dataset with 41 features
Reduced Dataset with six important features
Data Splitting
Training Set
Test Set
Successful Classification
Intrusion Classification
Gated Recurrent Unit Classifier
Training Phase
Satisfied Detection Rate
Testing Phase
Un-satisfied Classfication Results
Un-satisfied Detection Rate

Figure 9: End to End data flow of the machine learning model

4.6 Experimental Settings:

4.6.1 Data Preparation:

As explained above we have selected the 10% KDD dataset to train and test the machine learning algorithms. The dataset is split into different layers based on the attack types at the respective TCP/IP layers as shown in the architecture diagram. We are not considering Link Layer as part of this research because no attack type in the data set falls into the Link Layer category. All attack types in the dataset fall under one of the three TCP/IP layer category shown in below Table 1:

Layer	Attack Types
Application Layer	pod,Smurf,back,buffer_overflow,load module, warezmaster,Perl,Nmap, guess_passwd,Satan,impa,ftp_write, multihop and normal.
Transport Layer	Neptune, land, teardrop, port sweep, buffer overflow, Nmap and normal
Network Layer	overflow, Smurf, pod, IP sweep and normal
All Layer IDS	Normal and all attack types

Table 1: Attack Types for each TCP/IP Layer

Each sample is read and appended to a new dataset based on the attack type. The dataset for the Application Layer IDS has 382,266 samples, out of which 97,278 samples are "normal" and the remaining samples are categorized as one of the attack types. The dataset for the Transport Layer has 206,780 samples, out of which 109,502 are categorized as one of the attack types and the remaining samples are "normal". The dataset for the Network layer and all layers contains 379,609 and 490,251 samples respectively. The dataset contains three categorical features which need to be encoded into numerical form before they are provided as input to the algorithm model. The features "protocol_type", "service" and "flag" are encoded to numerical values. For every data set, 80% of the data is considered as the training data and 20% of the data is considered as the testing data. Each dataset is later divided into a features set and the corresponding label set. We have encoded the label "normal" as [0 1] and "all other attack types" as [1 0].

4.6.2 Feature Engineering

The main motive of this research is to build a light-weight security solution for IoT systems. Therefore, it is important to reduce the number of features and use only the important features required for training and testing the algorithm. We have used a random forest classifier as the feature selection technique which is proven as the best method for reducing the dimensionality for the KDD'99 Cup dataset [12]. The random forest uses tree-based methods that rank the features importance based on their ability to improve the node purity (Gini impurity). We have graphically visualized the importance of each feature and selected the top six features for each dataset before inputting them to the model. The decrease in the features in the input data from 41 to 6 makes the model faster to train and respond, making it flexible and adaptive. The graphical visualizations and the selected feature set for each IDS are detailed in the next chapter

4.6.3 Hardware and Software used

We have selected Google's Tensorflow to perform the experiments as it provides an option to visualize the network design which is important for the developers. The experiments are implemented in the below environment:

CPU: Intel ® Core ™ i5-5200U CPU @ 2.20 GHz
RAM: 8GB
OS: Windows 10
Programming Language: Python
Libraries used: numpy, scikit-learn, pandas, and Tensorflow.

5. Results:

5.1 Introduction:

In this chapter, we provide the detailed results for each IDS classifier obtained using a Gated-Recurrent-Unit (GRU) neural network and their evaluation measures. We started the experiments with a light-weight GRU where we used one hidden layer and one hidden unit. We performed 10 sets of experiments for each hyper parameter set (learning rate, time-steps, hidden layers) and tuned them to obtain the optimized results. This turned out to be a binary classification problem where the classifier classifies each sample as “normal” or “attack”. Hence, we used the evaluation metrics for classification like accuracy, precision, recall, false positive rate and AUC. We have decided upon the best models by considering every relevant metric for the defined research problem which will be explained in detail in the later part of the chapter.

5.2 Feature Selection

As explained in previous chapters, we have used the random forest classifier algorithm to select the top important features for each IDS classifier. The selected features and the graphical results of the importance of the features for each classifier are shown in Table 2 and Figures 10,11,12,13. It can be seen from the table that “Protocol_type” feature has been selected in all the intrusion detection layers. This shows that the feature “Protocol_type” provides high information to categorize the label as “attack” or “normal”. It can also be seen that the application layer IDS and network layer IDS select four features (Protocol_type, count, srv_count, dst_host_count) in common.

Layer Type	Features Selected
All layers	Protocol_type, service, flag, src_bytes, dst_bytes, logged_in, count, srv_count, same_srv_rate, diff_srv_rate, dst_host_same_srv_rate, dst_host_diff_srv_rate
Application Layer	Protocol_type, flag, count, srv_count, dst_host_count, dst_host_same_src_port_rate
Network Layer	Protocol_type, src_bytes, count, srv_count, dst_host_count, dst_host_same_srv_rate
Transport Layer	Service, count, srv_error_rate,same_srv_rate, diff_srv_rate, dst_host_same_src_port_rate

Table 2: Selected Features list for each IDS classifier based on the performance

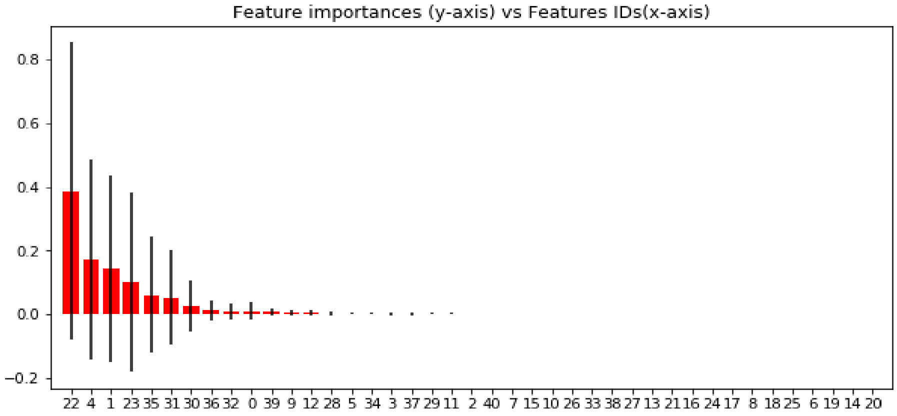


Figure 10: Feature Importance graph for all layers IDS

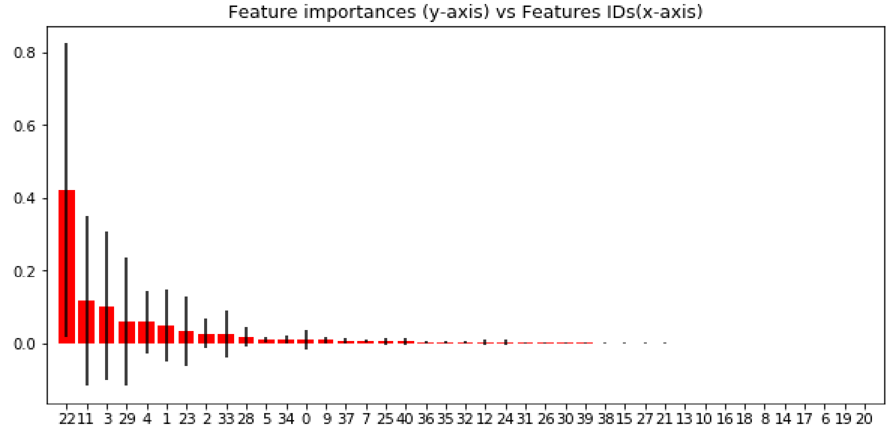


Figure 11: Feature Importance graph for Application Layer IDS

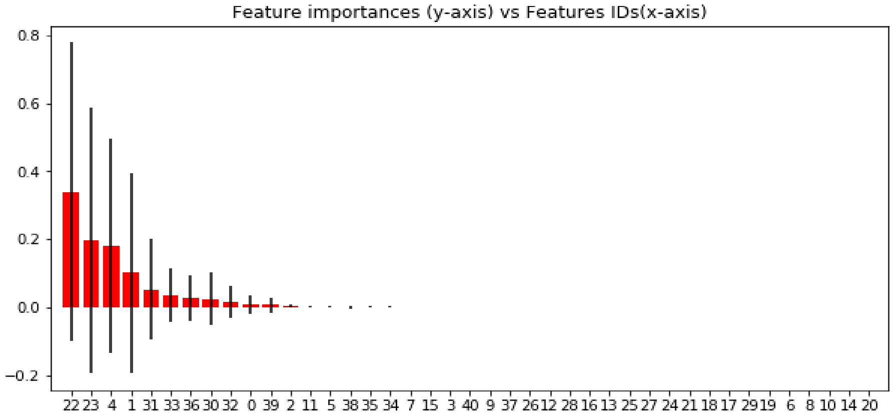


Figure 12: Feature Importance graph for Network Layer IDS

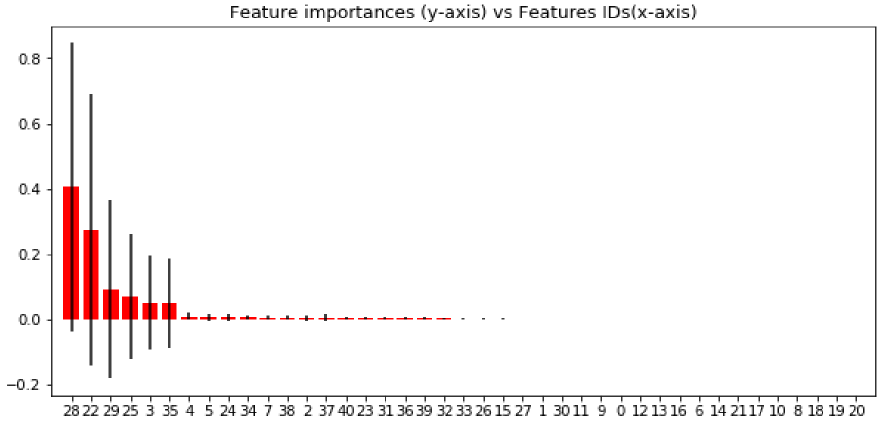


Figure 13: Feature Importance graph for Transport Layer IDS

5.3 Evaluation Metrics

In this section, we have evaluated the performance of the IDS classifiers by tuning the hyper-parameters of the GRU algorithm. We have performed a similar type of experiments on each IDS classifier (All layers, application layer, transport layer, and network layer classifiers). We compared the values of training accuracy, recall and false alarm rate with learning rate and time-steps to understand the behavior of model with change in hyper-parameters.

5.3.1 Performance results of All-Layer IDS Classifier:

In this experiment section, first, we started with a time-step range [10,20,30,40,50,60,70,80,100] and selected the time-steps which has best training accuracy. After selecting the time-step we searched for the learning-rate which produces best training accuracy. We used one hidden layer and one hidden unit in the network to satisfy the “light-weight” property of the IoT system. The detailed results are shown in Table 3.

Time-Steps	Train Accuracy	Precision	Recall	F-1 Score	FAR
10	98.694	0.9994	0.99	0.9977	0.0022
20	98.833	0.9943	0.3296	0.9922	0.0077
30	96.71	0.986	0.9952	0.9418	0.05812
40	98.706	0.9996	0.9902	0.9983	0.0016
50	97.7799	0.9967	0.9919	0.9865	0.0134
60	84.89	0.8914	0.9935	0.5011	0.4988
70	**98.911**	**0.9981**	**0.9939**	**0.9923**	**0.0076**
80	98.359	0.9999	0.9842	0.9997	0.0002
90	98.382	0.9995	0.9859	0.9981	0.0018
100	97.618	0.9937	0.9925	0.974	0.0257

Table 3: Evaluation Metrics for All Layer IDS Classifier

From the Table above, it can be inferred that the model performance is optimized when the input is given with ‘70’ time-steps and thus, this value is selected for further experiments for the All-Layers IDS in the research. The plots for the impact of time-steps and learning rate over recall, false alarm rate and training accuracy for All-Layer IDS classifier can be observed in Figures 14,15,16.

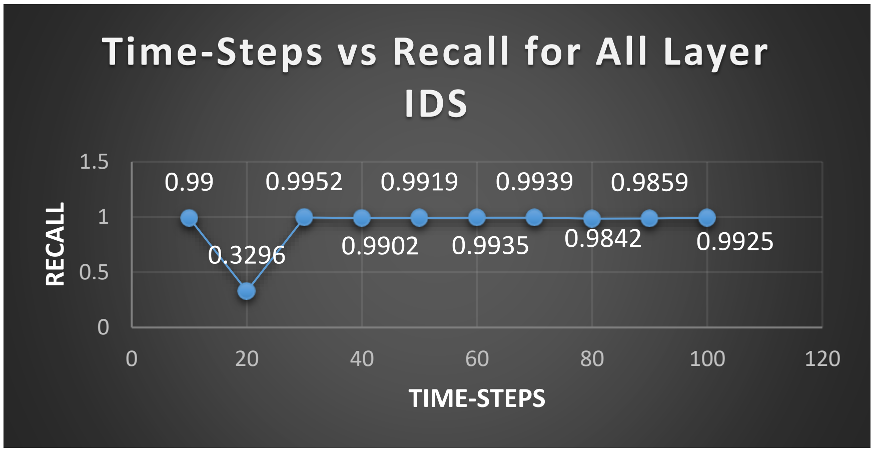


Figure 14: Impact of time-steps on recall in All-Layer IDS classifier

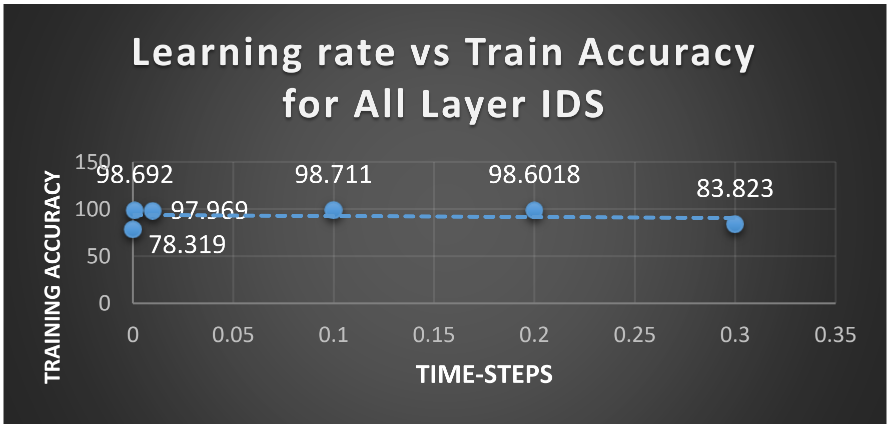


Figure 15: Impact of learning rate over training accuracy in All-Layer IDS classifier

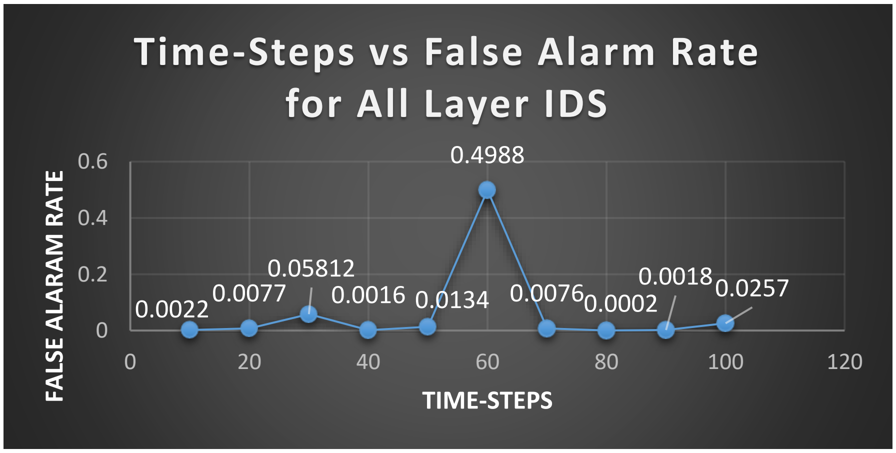


Figure 16: Impact of time-steps on false alarm rate (FAR) in All-Layer IDS classifier

The confusion matrix in Figure 17 provides information about the true-negatives (TN), false-negatives (FN), false-positives (FP) and true-positives (TP). The detailed confusion matrix values for the All-Layer IDS classifier with the best hyper-parameter combination (time-steps = 40, learning_rate = 0.01) is presented in Table 4 and Figure 17:

True-Negatives	76503
False-Positives	1467
False-Negatives	8394
True-Positives	308813

Table 4: Confusion Matrix for All-Layer IDS

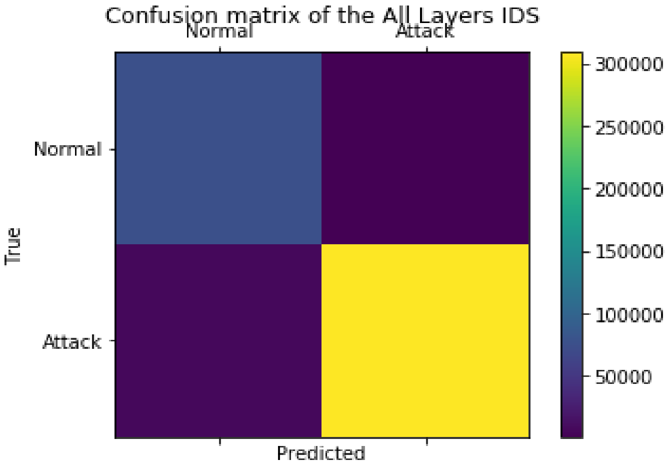


Figure 17: Confusion Matrix plot for the All-Layer IDS

5.3.2 Performance results of Application Layer IDS classifier

In this experiments section, we have used the dataset with attacks occurring at the application layer which was prepared during the data preparation stage. A similar series of experiments as in the All-Layer IDS case is performed on the Application Layer IDS and we have achieved the best training accuracy with '40' time steps. The complete results for Application Layer IDS can be interpreted in Table 5 and Figure 18, 19, 20.

Time-Steps	**Train Accuracy**	**Precision**	**Recall**	**F-1 Score**	**FAR**
10	93.73	0.9242	0.9978	0.9596	0.2414
20	99.71	0.9989	0.9971	0.998	0.003
30	96.6	0.9678	0.9871	0.9774	0.0964
40	**99.72**	**0.9998**	**0.9965**	**0.9981**	**0.0005**

50	99.04	0.9998	0.9837	0.9935	0.0003
60	99.6	0.9999	0.9948	0.997	0.0001
70	98.05	0.99104	0.9827	0.9868	0.026
80	99.671	0.9998	0.9937	0.9968	0.0003
90	98.767	0.9969	0.9864	0.9917	0.0088
100	99.64	0.9996	0.9956	0.997	0.0011

Table 5: Evaluation Metrics for Application Layer IDS classifier

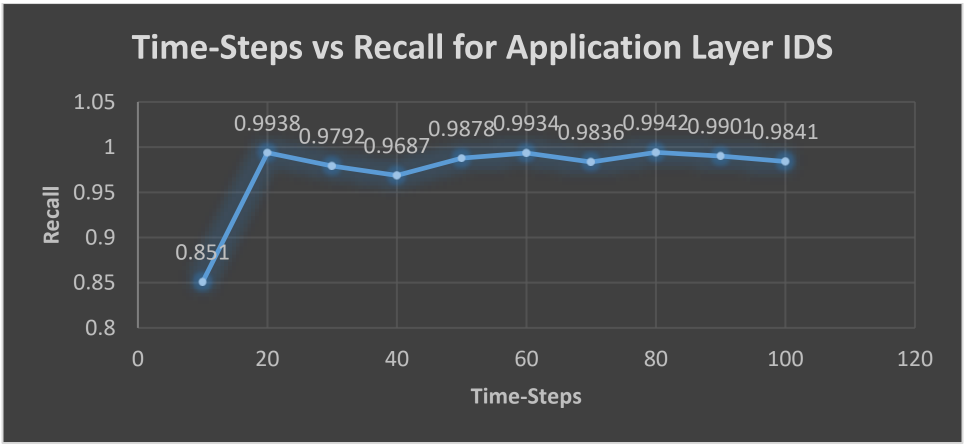


Figure 18: Impact of time-steps on recall in Application-Layer IDS classifier

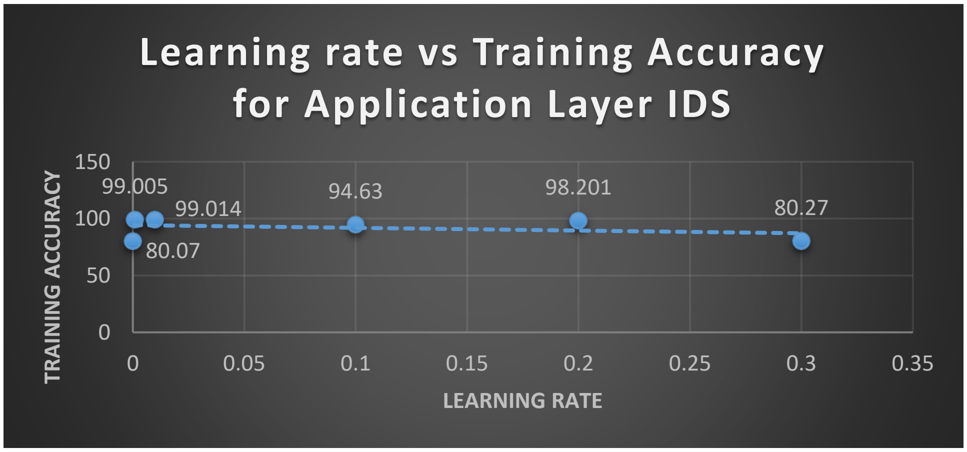


Figure 19: Impact of learning rate over training accuracy in Application-Layer IDS classifier

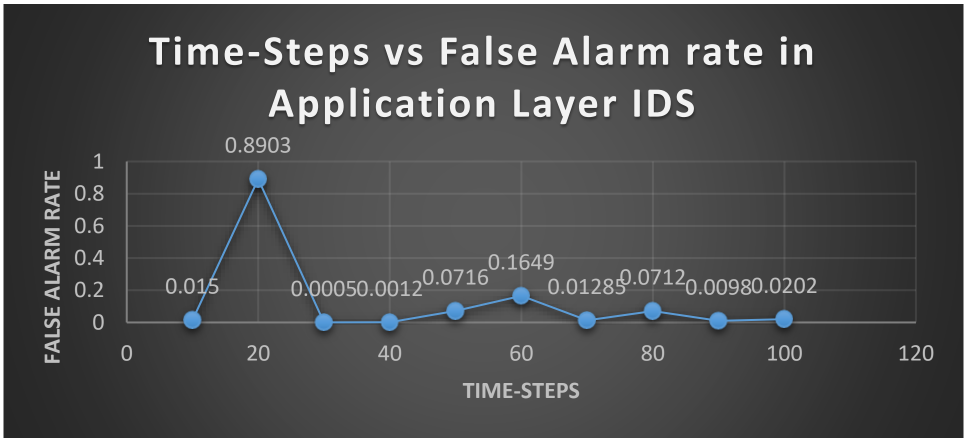


Figure 20: Impact of time-steps on false alarm rate (FAR) in Application-Layer IDS classifier

The confusion matrix of the optimized (time-steps = 40, learning rate – 0.01) and the corresponding plot for the Application-Layer IDS can be analyzed in Table 6 and Figure 21.

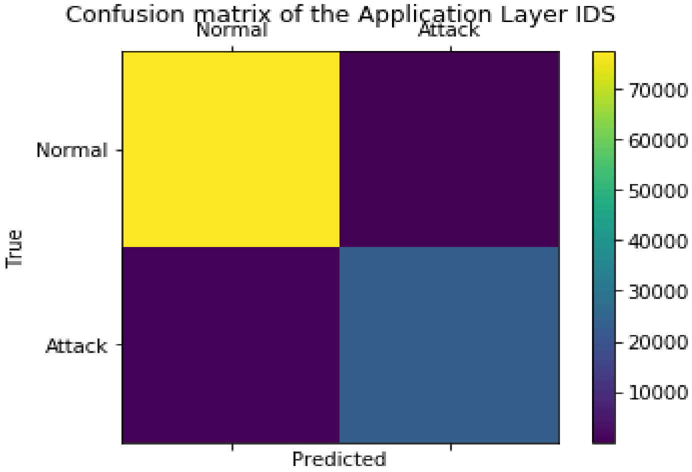


Figure 21: Confusion Matrix plot for the Application-Layer IDS

True-Negatives	77305
False-Positives	44
False-Negatives	795
True-Positives	22768

Table 6: Confusion Matrix for Application-Layer IDS

5.3.3 Performance results of Transport-Layer IDS classifier:

The set of experiments for the Transport-Layer IDS classifier can be found in Table 7, Figures 22, 23, 24. We have achieved best-optimized results with time-steps '60' when processed with learning rate 0.01. All the experiments are performed on the transport layer data set which contains those attacks of samples belonging to the transport layer with results presented in Table 7 and Figures 22,23 24.

Time-Steps	**Train Accuracy**	**Precision**	**Recall**	**F1 Score**	**FAR**
10	99.3	0.9938	0.993	0.9934	0.0068
20	98.78	0.9952	0.981	0.9884	0.0052
30	99.27	0.9995	0.9867	0.9931	0.0005
40	99.05	0.9988	0.9832	0.9909	0.0012
50	99.106	0.998	0.985	0.9914	0.0022
60	**99.475**	**0.9988**	**0.9911**	**0.995**	**0.0012**
70	99.43	0.9994	0.9902	0.9948	0.0005
80	98.83	0.9986	0.9793	0.9888	0.0015
90	98.74	0.9977	0.9785	0.988	0.0024
100	99.167	0.9963	0.9878	0.992	0.004

Table 7: Evaluation Metrics for Transport Layer IDS classifier

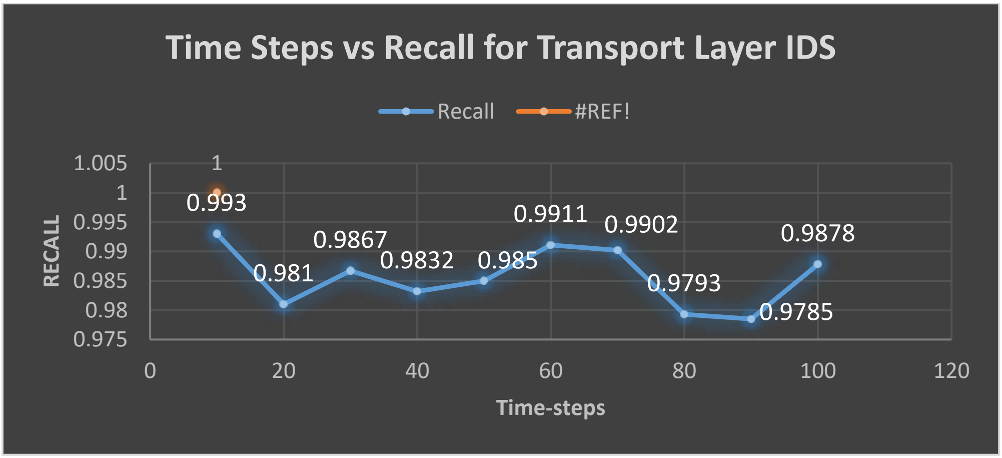


Figure 22: Impact of time-steps on recall in Transport-Layer IDS classifier

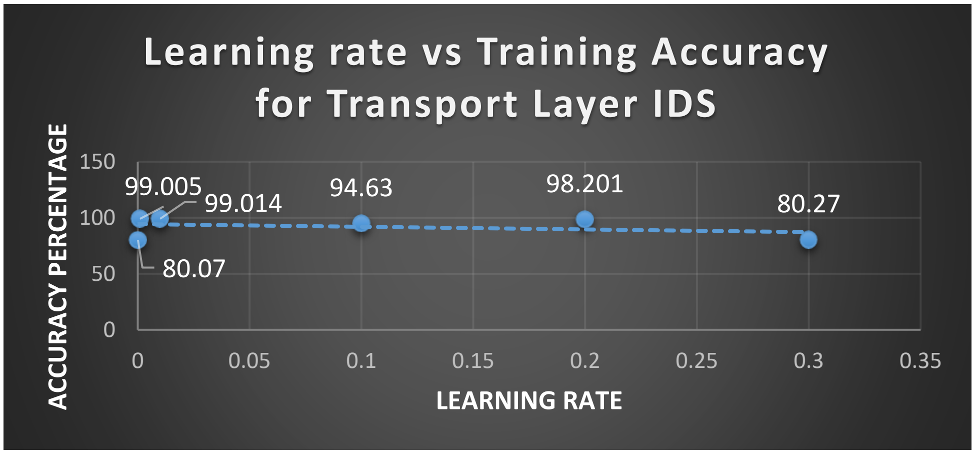


Figure 23: Impact of learning rate over training accuracy in Transport-Layer IDS classifier

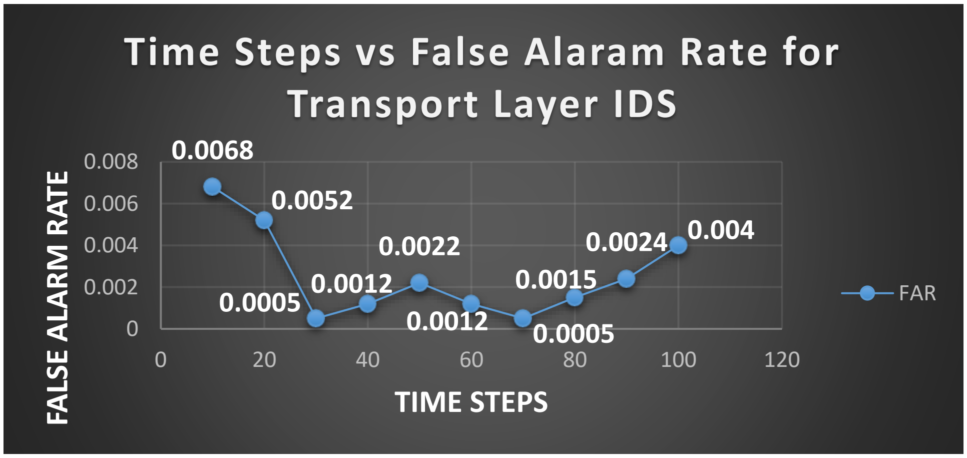
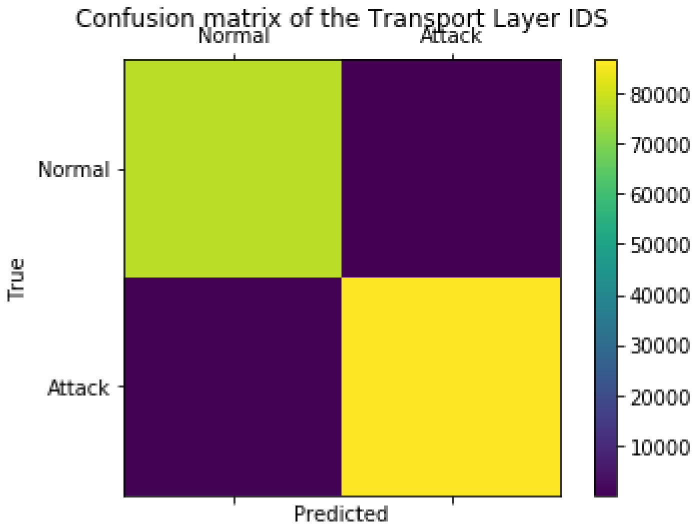


Figure:24 Impact of time-steps on false alarm rate (FAR) in Transport-Layer IDS classifier

The confusion matrix of the optimized (time-steps = 60, learning rate – 0.1) and the corresponding plot for the Transport-Layer IDS can be analyzed in Table 8 and Figure 25.



True-Negatives	77853
False-Positives	97
False-Negatives	771
True-Positives	86643

Figure 25: Confusion Matrix plot for the Transport-Layer IDS

Table 8: Confusion Matrix for Transport-Layer IDS

5.3.4 Performance results of Network-Layer IDS classifier:

The results for the Network IDS classifier are detailed in Table 9, Figures 26,27,28. It can be observed that the best-optimized results are obtained for the time-steps ‘40' and learning rate ‘0.001'.

Time-Steps	Train Accuracy	Precision	Recall	F1 score	FAR
10	99.32	0.9993	0.9915	0.9954	0.0018
20	99.21	0.9999	0.9894	0.9964	0.00006
30	99.42	0.9989	0.9932	0.9961	0.0029
40	**99.901**	**0.9921**	**0.9938**	**0.993**	**0.018**
50	99.49	0.9998	0.993	0.9964	0.0002
60	99.45	0.9999	0.9927	0.9963	0.00006
70	99.43	0.9943	0.9999	0.9961	0.00026
80	99.42	0.9999	0.9922	0.996	0.0012
90	99.45	0.9999	0.9927	0.9963	0.0001
100	99.39	0.9977	0.9941	0.9959	0.0064

Table 9: Evaluation Metrics for Network Layer IDS classifier

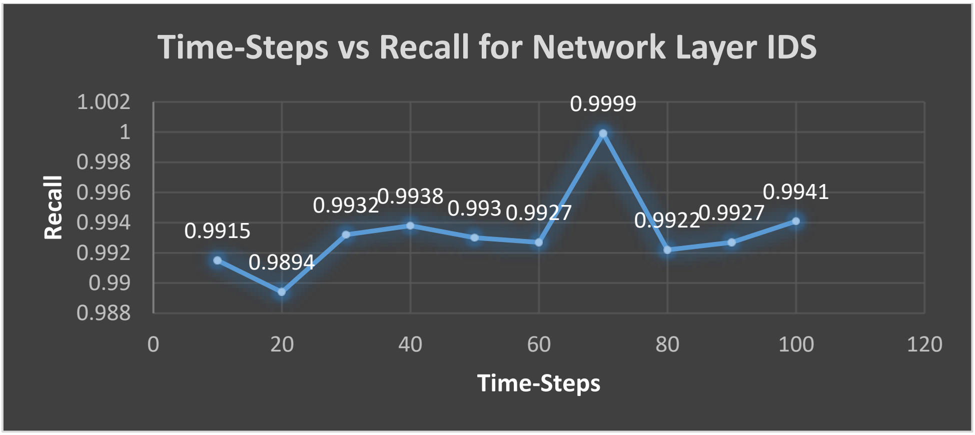


Figure 26: Impact of time-steps on recall in Network-Layer IDS classifier

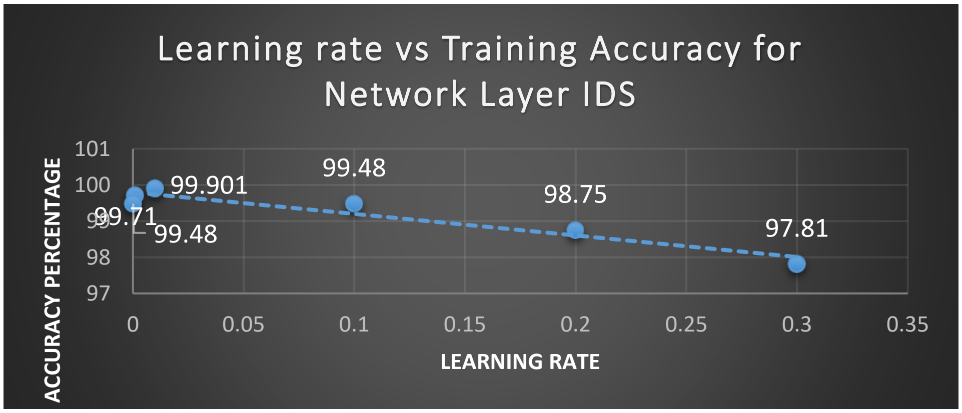


Figure 27: Impact of learning rate over training accuracy in Network-Layer IDS classifier

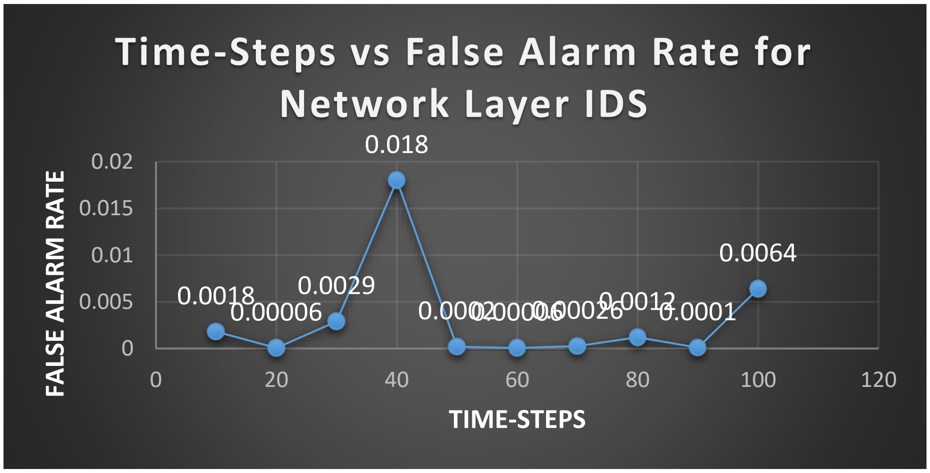


Figure 28: Impact of time-steps on false alarm rate (FAR) in Network-Layer IDS classifier

The confusion matrix of the optimized (time-steps = 40, learning rate – 0.001) and the corresponding plot for the Transport-Layer IDS can be analyzed in Table 10 and Figure 29.

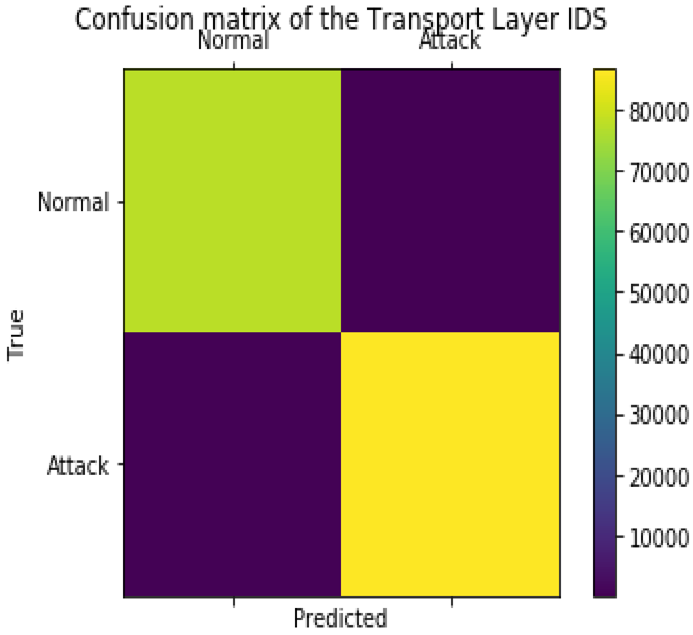


Figure 29: Confusion Matrix plot for the Network-Layer IDS

True-Negatives	77865
False-Positives	43
False-Negatives	1510
True-Positives	224229

Table 10: Confusion Matrix for Network-Layer IDS

5.4 Comparison of the results for IDS classifiers:

The optimized results of all the IDS classifiers are compared and it was found that the performance of the All-Layers IDS classifiers is inferior to the individual layer IDS classifiers in terms of training accuracy and training time. The light-weight algorithms, when used in a multi-layer architecture, perform better which is suitable for an IoT system. The comparison of the results can be found in Table 11:

Feature Selection Method	IDS Type	Number of features	Training Accuracy	Training Time
Random Forest Classifier	All Layer IDS	12	98.911	58.64 seconds
	Application Layer IDS	6	99.72	35.84 seconds
	Transport Layer IDS	6	99.475	27.91 seconds
	Network Layer IDS	6	99.71	31.03 seconds

Table 11: Performance comparison among proposed IDS classifiers

5.5 Comparison of the IDS classifiers performance with existing work

We have performed additional analysis and compared the results with existing research performed by machine learning algorithms on intrusion detection classification as seen in Table 12. It can be observed that our research has outperformed the performances of all the existing work.

Algorithm	Precision (%)	Recall (%)	Accuracy (%)	FAR (%)
GNNN [22]	87.08	59.12	93.05	12.46
FNN [22]	92.47	86.89	97.35	2.65
RBNN [22]	69.56	69.83	93.05	6.95
Jordan ANN [23]	-	62.9	-	37.09
KNN [25]	-	91	-	8
K-Mean-KNN [27]	98	98.68	93.55	47.9
LSTM RNN [24]	-	98.88	96.93	10.04
Fuzzy association rule [26]	-	91	-	3.34
RNN Hessian-free [28]	-	95.37	-	2.1
GRU RNN [20]	95.72	98.65	97.06	10.01
All Layers IDS	**99.81**	**99.39**	**98.91**	**0.76**
Application Layers IDS	**99.98**	**99.65**	**99.72**	**0.05**
Transport Layer IDS	**99.88**	**99.11**	**99.47**	**0.12**
Network Layer IDS	**99.21**	**99.38**	**99.9**	**1.8**

Table 12: Comparisons of existing IDS classifiers to the proposed IDS classifiers

6. Conclusion and Future Work

This inter-disciplinary research is novel in a way that, for the first time, it has applied deep learning methods for IoT security. We performed a rigorous analysis of the architecture of IoT, followed by its security issues and privacy issues. As part of this research, we limited the scope of this thesis to network data security. We have proposed a light-weight architecture for an Intrusion Detection System (IDS) in an IoT network. Based on TCP/IP layer architecture and the attack types at each layer, we have suggested placing IDS classifiers at each layer. This has reduced the data set size at each classifier and improved the performance in terms of accuracy, recall, training time and false alarm rate. We have applied deep learning algorithms to classify the data at each IDS classifier. This approach has achieved outstanding results with better results than existing work in the literature. Moreover, we have used the full KDD 99'cup 21% data set for the experiments, unlike previous research work. As seen in section 4.4, the training time of Transport Layer IDS, Application Layer IDS and Network Layer IDS is almost half of the All Layer IDS which is important for dynamic IoT networks. As seen in section 4.5, the accuracy and false alarm rate of All-Layer IDS is 98.91% and 0.76% respectively which outperformed all other existing IDS classifiers in literature.

The applications of deep learning to IoT applications to develop security solutions is still in the naïve stage, and we believe, it has a lot of potentials. As the IoT deals with user's personal data and industry's information, it is crucial to implement robust solutions to protect from security threats. This can be possible with the concepts of machine learning and deep learning as IoT generate a humongous amount of heterogeneous data. We have applied Gated-Recurrent-Unit neural networks to the dataset. However, there are many improvised versions of recurrent neural networks such as Dynamic RNN, Bi-Directional RNN which can achieve better performances than basic GRU cells. One can also build a hybrid network using convolutional neural networks and recurrent neural networks to deal with multi-modal data. This research focused on dealing with IoT devices where the processing power is low and the data size is not huge. This research can be taken forward by applying it to large amounts of real-time IoT data.

The Internet of Things (IoT) is a revolution rather than an evolution. As the IoT evolves, the security issues evolve. The IoT is a boon to the society only when it is secured, and this can possible with artificial intelligence.

BIBLIOGRAPHY

[1] Farooq, M. U., Waseem, M., Khairi, A., & Mazhar, S. (2015). A critical analysis on the security concerns of the internet of things (IoT). *International Journal of Computer Applications*, *111*(7).

[2] Dua, S., & Du, X. (2016). *Data mining and machine learning in cybersecurity*. CRC press.

[3] Alpaydin, E. (2014). Introduction to machine learning. MIT press

[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444

[5] K. Greff, et al., "LSTM: A Search Space Odyssey", arXiv preprint arXiv: 1503. 04069, 2015.

[6] Murata, N., Yoshizawa, S., & Amari, S. I. (1994). Network information criterion-determining the number of hidden units for an artificial neural network model. IEEE Transactions on Neural Networks, 5(6), 865-872

[7] Xu, K., Wang, X., Wei, W., Song, H., & Mao, B. (2016). Toward software defined smart home. IEEE Communications Magazine, 54(5), 116-122

[8] Pan, G., Qi, G., Zhang, W., Li, S., Wu, Z., & Yang, L. T. (2013). Trace analysis and mining for smart cities: issues, methods, and applications. *IEEE Communications Magazine*, *51*(6), 120-126.

[9] Luo, X., Liu, J., Zhang, D., & Chang, X. (2016). A large-scale web QoS prediction scheme for the Industrial Internet of Things based on a kernel machine learning algorithm. *Computer Networks*, *101*, 81-89.

[10] Esmalifalak, M., Liu, L., Nguyen, N., Zheng, R., & Han, Z. (2014). Detecting stealthy false data injection using machine learning in smart grid. *IEEE Systems Journal*.

[11] Abie, H., & Balasingham, I. (2012, February). Risk-based adaptive security for smart IoT in eHealth. In *Proceedings of the 7th International Conference on Body Area Networks* (pp. 269-275). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[12] Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature Selection for Intrusion Detection Using Random Forest. *Journal of Information Security*, *7*(03), 129.

[13] Li, Y., & Guo, L. (2007). An active learning based TCM-KNN algorithm for supervised network intrusion detection. *Computers & security*, *26*(7), 459-467.

[14] Leung, K., & Leckie, C. (2005, January). Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38* (pp. 333-342). Australian Computer Society, Inc..

[15] Ashfaq, R. A. R., Wang, X. Z., Huang, J. Z., Abbas, H., & He, Y. L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, *378*, 484-497.

6] Tao, X., Kong, D., Wei, Y., & Wang, Y. (2016). A Big Network Traffic Data Fusion Approach Based on Fisher and Deep Auto-Encoder. *Information*, *7*(2), 20.

[17] Kim, J., & Kim, H. (2017, February). An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization. In *Platform Technology and Service (PlatCon), 2017 International Conference on* (pp. 1-6). IEEE.

[18] Staudemeyer, R. C., & Omlin, C. W. (2013, October). Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data. In *Proceedings of the*

South African Institute for Computer Scientists and Information Technologists Conference (pp. 218-224). ACM.

[19] Staudemeyer, R. C. (2015). Applying long short-term memory recurrent neural networks to intrusion detection. *South African Computer Journal*, *56*(1), 136-154.

[20] Kim, J., Kim, J., & Kim, H. (2015). An Approach to Build an Efficient Intrusion Detection Classifier. *JOURNAL OF PLATFORM TECHNOLOGY*, *3*(4), 43-52.

[21] Breiman, L., 2001. Random forests. Machine Learning 45 (1), 5–32.

[22] Devaraju, S., & Ramakrishnan, S. (2014). PERFORMANCE COMPARISON FOR INTRUSION DETECTION SYSTEM USING NEURAL NETWORK WITH KDD DATASET. *ICTACT Journal on Soft Computing*, *4*(3).

[23] Beghdad, R. (2007). Training all the KDD data set to classify and detect attacks. *Neural Network World*, *17*(2), 81.

[24] Kim, J., Kim, J., Thu, H. L. T., & Kim, H. (2016, February). Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. In *Platform Technology and Service (PlatCon), 2016 International Conference on* (pp. 1-5). IEEE.

[25] Han, S. J., & Cho, S. B. (2003). Detecting intrusion with rule-based integration of multiple models. *Computers & Security*, *22*(7), 613-623.

[26] Tajbakhsh, A., Rahmati, M., & Mirzaei, A. (2009). Intrusion detection using fuzzy association rules. *Applied Soft Computing*, *9*(2), 462-469.

[27] Tsai, C. F., & Lin, C. Y. (2010). A triangle area based nearest neighbors' approach to intrusion detection. *Pattern recognition*, *43*(1), 222-229.

[28] Kim, J., & Kim, H. (2015, August). Applying Recurrent Neural Network to Intrusion Detection with Hessian Free Optimization. In *International Workshop on Information Security Applications* (pp. 357-369). Springer, Cham.

[29] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html