Deep Learning-Driven Heuristic Intrusion Detection System

Minor Project Report

Submitted in partial fulfillment of the requirement for the registration of the degree of

Bachelor of Technology

in

COMPUTER ENGINEERING

by

Abhishek Panda 2K21/CO/19

Pranay Avnish Kachhap 2K21/CO/338

Pritesh Das 2K21/CO/348

under the supervision of

Dr. Nipun Bansal

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

DEPT. OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

CERTIFICATE

This is to certify that the report entitled **Deep Learning-Driven Heuristic Intrusion Detection System** submitted by Abhishek Panda, Pranay Avnish Kachhap, Pritesh Das (2K21/CO/19, 2K21/CO/338, 2K21/CO/348) in partial fulfilment of the B.Tech., Delhi Technological University. A degree in Computer Engineering which is an authentic record of the work carried out by them under our guidance, supervision and care. This report in any form has not been submitted to any other University or Institute for any purpose.

Place : Delhi
Date : 12th December 2024

Dr. Nipun Bansal
Project Supervisor

DEPT. OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

DECLARATION

We hereby certify that the project report, Deep Learning-Driven Heuristic Intrusion Detection System, submitted to Delhi Technological University, Delhi, as a partial fulfilment of the requirements for the award of a Bachelor of Technology degree, is an authentic work completed by us under Dr. Nipun Bansal's supervision. This submission reflects our thoughts in our own words, and where we have borrowed from others, we have properly and appropriately cited and referenced the original works.

Place : Delhi **Abhishek Panda (2K21/CO/19)**
Date : 12 December 2024 **Pranay Kachhap (2K21/CO/338)**
Pritesh Das (2K21/CO/348)

DEPT. OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

ACKNOWLEDGEMENT

We, Abhishek Panda (2K21/CO/19), Pranay Kachhap (2K21/CO/338) and Pritesh Das (2K21/CO/348), take this opportunity to express our deepest sense of gratitude and sincere thanks to everyone who helped us to complete this work successfully. We express our sincere thanks to Dr. Vinod Kumar, Head of Department, Department of Computer Engineering, Delhi Technological University, Delhi for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to Dr. Nipun Bansal, Department of Computer Engineering, Delhi Technological University for the support and cooperation alongside his really beneficial Mentorship and Knowledge.

Place : Delhi **Abhishek Panda (2K21/CO/19)**
Date : 12 December 2024 **Pranay Kachhap (2K21/CO/338)**
Pritesh Das (2K21/CO/348)

ABSTRACT

In the ever changing domain of cybersecurity, the rise of complex cyber threats necessitates the development of versatile defense mechanisms. This paper explores a Deep Learning-Driven Heuristic Intrusion Detection System (DL-HIDS) developed to identify and mitigate complex attack vectors in real-time. Leveraging the capabilities of deep neural networks, the proposed system combines heuristic analysis with deep learning techniques to identify anomalous patterns and potential intrusions in network traffic.

The heuristic component of DL-HIDS employs rules based strategies and behavior analysis to identify suspicious activities that may indicate potential threats. These heuristics are dynamically modified to adapt to emerging attack patterns, improving the system's ability to address zero-day vulnerabilities. Meanwhile, the deep learning module utilizes advanced architectures such as convolutional neural networks (CNNs[6][10]) and recurrent neural networks (RNNs) to process vast amounts of data and uncover hidden patterns that traditional methods might overlook. By integrating these components, DL-HIDS attains a balance between predefined rules and the adaptive learning capabilities of neural networks.

We evaluate the system using benchmark datasets such as InSDN well as real-world traffic scenarios to simulate a wide range of intrusion attempts, including denial-of-service (DoS) attacks, ransomware, and advanced persistent threats (APTs). The experimental results indicate that DL-HIDS outperforms traditional intrusion detection systems in terms of detection accuracy, precision, recall, and false-positive rates. Additionally, the system's computational efficiency and scalability make it well capable for deployment in large-scale and high-speed network environments.

This research not only highlights the effectiveness of combining heuristics with deep learning in intrusion detection but also emphasizes the importance of real-time adaptability and robustness in modern cybersecurity solutions. By addressing the limitations of existing systems, DL-HIDS provides a comprehensive framework for protecting critical digital infrastructures against the dynamic landscape of cyber threats.

CONTENTS

LIST OF FIGURES

INTRODUCTION

With the rise of internet usage, we have seen significant advancements in various industries, but have also introduced another problem: Network Security. One solution to this is Intrusion Detection Systems (IDS). They have become a critical component in protecting networks against attackers. Traditional IDS relies on predefined signatures and rules, which worked for a while. However they fail to detect 0 day exploits, and fail to adapt to the ever evolving nature of cybersecurity. This calls for a need to implement IDS using Machine Learning techniques to enhance the capabilities of IDS.

Machine learning based Intrusion Detection Systems (IDSs) are able to analyze large amounts of network data to identify patterns that imply malicious activities. These systems are able to adapt to new forms of attacks by learning from previous data, making them more efficient compared to traditional systems. A wide variety of Machine learning methods, inclusive of, but not limited to such as: deep learning, support vector machines (SVM), and ensemble methods, have been utilized to improve the overall accuracy of detection and lower false alarm rates in IDS.

Despite the promising results, several challenges remain in the implementation of ML-based IDS. One major problem is inability in detecting low-frequency attacks which are ignored due to the presence of commonly available attack patterns in the training data. Furthermore, the changing property of cyber attacks makes it necessary to keep updating the datasets that are used for model training while ensuring they are effective in detecting new and upcoming attack vectors. Additionally ML models make it difficult for cybersecurity experts to trust the decisions made by these systems, thus making it important to have simple and easy to explain AI frameworks.

To conclude, while machine learning based IDS provides several advantages over traditional approaches, we need to address the challenges faced by traditional IDS systems. By doing so, we end up with more robust and reliable solutions to protect our digital infrastructure against the ever-evolving landscape of cyber attacks.

MOTIVATION

Deep Learning has the capacity to analyze enormous amounts of data and recognize patterns that machine learning based approaches can miss. Our solution could potentially address the concern of traditional heuristic based IDS, improve its accuracy, reduce the false positive rates, limited adaptability and its resilience against cyber attacks. The main motivation behind this paper is the urgent need to solve this problem of traditional IDS systems. We would also like to leverage the concepts of software defined networking, as most modern day IOT systems implement software defined networking to some extent.

Software defined approaches has a lot of potential for analyzing large amounts of data and detecting recognizable patterns that traditional approaches can miss, deep learning can be a viable alternative. The suggested DL-HIDS addresses the limitations of conventional systems, which includes high false positive rates and limited adaptability, through aggregating heuristics methods with deep learning. The urgency for improving the accuracy and reliability of IDS for securing important infrastructures in this world that's interconnected digitally is what fuels this study.

DL-HIDS uses behavioral analysis and rule-based tactics to find suspicious activity. The system's ability to learn and detect zero day vulnerabilities is improved by these heuristics that dynamically change to adapt to new attack patterns. In the meantime, the deep learning module processes tons of data to find underlying patterns that traditional approaches could never find by utilizing sophisticated algorithms like RNNs and LSTMs[4]. By combining these elements, DL-HIDS strikes a good balance between signature based rules and neural network's ability to learn.

To replicate diverse forms of intrusion attempts, like ransomware, advanced persistent threats(APTs), and denial-of-service(DoS) attacks, we use benchmark datasets such as InSDN, Orion, UTSA, and Mendely Data[1] along with real-world traffic scenarios. According to previous findings, DL-HIDS is a better performer than conventional intrusion detection systems when considering parameters such as positive rates, recall, detection accuracy, and precision.

Furthermore, this system works well with large-scale, high speed networks, due its computational efficiency and the ability to scale the system. This study also underlines the importance of real-time flexibility and resilience to cyberattacks. DL-HIDS provides a robust framework for protecting digital data against the dynamic nature of cyberthreats by solving the issues of current solutions.

PROBLEM STATEMENT

Intrusion Detection Systems (IDS) that used signature based anomaly detection were accurate at detecting known vulnerabilities. However the serious flaw with it was the inability to detect zero day exploits, i.e exploits not used in the wild and is unknown to the security research community. This necessitated heuristic based intrusion detection techniques to resolve the issues with signature based anomaly detection systems.

Heuristic based intrusion detection system (IDS) were made to overcome the flaws of signature-based systems. By looking at the departures from normal behavior, these systems utilize anomaly detection and behavior-driven analysis methods to detect threats. Heuristic-based intrusion detection systems are capable enough in recognizing previously unknown threats, but may result in having high false rejection rates (FRR), which might identify neutral behavior as malicious, and false acceptance rates (FAR), which can incorrectly identified harmful activity as benign. The broad use of heuristic-based systems is hindered by a trade-off between detection and accuracy.

With the advent of DL-based intrusion detection systems (IDS) in recent years, intrusion detection has been greatly transformed beyond imagination. Solutions based on Deep Learning contribute to a major increase in detection rates for both known and unknown threats by utilizing highly efficient machine learning algorithms to find miniscule trends in complex datasets. These technologies offer strong opportunities and incentives to reduce cybersecurity threats and have exhibited a performance that surpasses initial expectations in a wide range of network situations.

It is in this context that our work becomes extremely helpful. Our research focuses on putting into practice and examining IDS solutions that make use of DL and is especially made for SDN[11] based infrastructures. To enhance the security of our network infrastructures, we want to minimize the gap between IDS techniques and SDN[11] technologies. Our strategy uses DL models to leverage the centralized control of SDNs[11] to guarantee that our system can identify and react to threats with very low latency and high precision. By addressing the significant gap in state of network security research, we want to set a paradigm for relatively safer SDN[11] utilizations.

LITERATURE REVIEW

6.1 Software Defined Networking:

Software-defined simplifies data communication in the network. It provides a method for network architecture to leverage software applications to provide more intelligent network control. The goal of software defined networks is to increase throughput of bandwidth and increase efficiency of the current network architecture.

In comparison to traditional network architecture, SDN[11] is way complex. The ethernet switches utilized in the current architecture are ordered in a tree-like fashion. The changing requirements of computing and storage are not met by current architecture. High speed, scalability and robustness are traits of the SDN[11].

When it comes to the Software-Defined Network model, the controller, which is a logical central entity, takes over network control decisions from the forwarding devices. It is only the controller that can establish and update control rules and network policies. These are then automatically downloaded into the switches and routers of the respective network infrastructure and configuration. A centralized controlling entity, easy programmability, and less complex network management are made more feasible through this method.

The forwarding devices responsible for sending incoming data to the desired location are represented by the data plane. The controller resides in the control plane. It collects traffic information from the forwarders and transmits control rules to them through the southbound API. The software required for network management is a part of the application plan. Using a very high level coding language, the administrators establish and automate the desired network policies with the help of the northbound API that is provided by the controller.

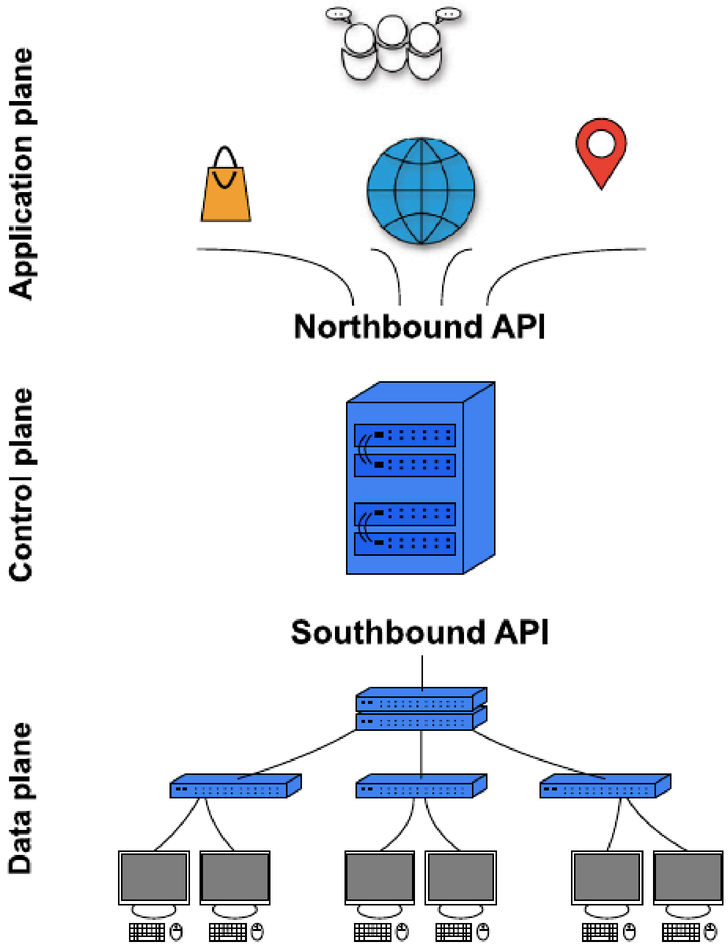

Fig 1: SDN architecture and network planes.

6.2 Deep Learning:

One of the subsets of machine learning is represented by deep learning. It is composed of mathematical models with multiple layers or levels of abstraction.Without the requirement for human involvement, these models are able to identify complex patterns in gigantic amounts of data because they essentially copy the structure of the human brain. Some very common examples of deep learning models are inclusive of the following:

6.2.1 Deep Neural Networks (DNNs[9]):

Deep Learning neural networks consist of several layers between the input and the output layer. An Artificial Neural Network is considered to be a deep neural network, if it consists of more than 3 layers and that includes the input and output layer. The "deepness" means how many hidden states there are in the neural network. Complex data relationships can be well expressed by the several hidden layers a DNN[9] can incorporate. This concept is used widely in Computer

Vision, Natural Language Processing, Simulation and Recommendation Systems, alongside Generative Models.

6.2.2 Convolutional Neural Networks (CNNs[6]):

Neural network that utilizes convolutional and pooling operations in the input to detect spatial patterns. Typically it requires less computational power than DNN[9] since its architecture reduces the amount of trainable parameter by utilizing lower neural connections

6.2.3 Auto Encoder (AE)[7]:

An AE[7] is a neural network developed for high precision compression and decompression of data. Its topology is very similar to that of a DNN[9]. It is made of two internal subnetworks, referred to as both a decoder and encoder which computes a low-dimensional representation of the supplied data. This is the responsibility of the decoder. The encoder is involved in the accurate reconstruction of the input which is made possible once it has been codified.

6.2.4 GAN[8] Model:

The GAN[8] model is made up of two inner networks with opposing training goals i.e. discriminator and generator. The discriminative binary classifier seeks to segregate samples with accuracy originating from within and out of its training set. The objective of generative is to generate artificial samples that nearly match the training set to trick the discriminator into falsely identifying them. The generator of a well trained GAN[8] model may produce genuine mimic samples that conform to training data's distribution.

6.2.5 Challenges:

None of the models mentioned can remember information about previously processed samples. This is a severe flaw, especially when our input data is sequential, where the order of the data matters. On the other hand, LSTM[4] & GRU[5], which are basically modified versions of RNNs (Recurrent neural networks) can remember long-term relationships between the inputs. LSTM[4] and GRUs[5] have neurons that have a feedback loop in order to achieve this. When processing new input data, these neurons can use information from previously processed data, thanks to this architecture. However these models require more processing power, especially because RNNs provide no ability to parallelly process input data.

6.3 Recurrent Neural Networks:

Recurrent neural networks (RNNs) implement past memory to have a relation between each step, leveraging the pattern of input data. RNNs have a hidden state that is updated at each time step as a sequence is processed. This makes RNNs suitable for problems where sequential data is used as input. For example, speech recognition, natural language processing, and time-series forecasting, where context or temporal dependencies are crucial. LSTMs[4] and GRUs[5] further improve on RNNs by using a gated cell, to address some issues with RNNs such as the exploding and vanishing gradient problem.

6.3.1 LSTM (Long Short-Term Memory):

Long term dependencies in sequential data are taken care of by LSTMs[4], one particular type of Recurrent Neural Networks (RNNs). LSTMs[4] achieve this by managing and adjusting the information flow with the help of a gating mechanism. The important components of LSTMs[4] consist of the following as mentioned below:

Forget Gate: This gate chooses which data to ignore, “forget” or discard.

Input Gate: The gate responsible for deciding which fresh data should be stored, “remembered” or retained.

Output Gate: The gate that regulates or manages the present step's output.

With a wide variety of applications such as text generation, speech recognition, and time-series forecasting, these gates allow LSTMs[4] to retain relevant and important data for very long periods of time.

6.3.2 GRU (Gated Recurrent Unit):

GRUs[5] are a more simplified version of LSTMs[4], often quicker and easier in comparison, that use less parameters to produce comparable or even better outcomes. They deploy a reset gate to manage memory and combine the input and forget gates into a single update gate. It consists of the following gates:

Update Gate: Achieves a proper balance between adding new information and retaining the old information.

Reset Gate: It makes decisions on how much of the past or previous information should be forgotten.

GRUs[5] are highly efficient in terms of computation and perform overwhelmingly well in tasks such as, but not limited to, detection of anomalies, language modeling, and captioning videos, especially when dealing with shorter sequences of data.

6.4 Intrusion Detection:

Malicious activities like denial of service attacks that are intentionally harmful are referred to as anomalies. Flash crowds and other acceptable aberrant instances are not contained in this definition. Any deviation from the statistical threshold observed in traffic can be regarded as an anomaly.

Effective detection of threats is important to protect computer networks and maintain services online, guaranteeing availability, security and integrity. Modern technologies that have been researched and created to overcome such risks are network intrusion detection systems.

These systems frequently gather and evaluate traffic data. On detection of suspicious behavior, the NIDS[2] sends an alarm, warning the administrators.

The two most commonly used approaches for NIDS[2] are anomaly-based and signature-based. Signatures based NIDS[2] uses a database of known attack signatures, and regularly needs to be maintained. Incoming traffic is analyzed, and any malicious activity matches with the signatures stored on the database, an abnormality is found and is reported to administrators. The main flaw with this approach is its inability to detect zero-day attacks or novel attacks.

Then comes the Anomaly-based approach, where we establish a secure baseline. If the incoming traffic deviates normal behaviour beyond a certain threshold, an intrusion is detected and triggers the alarm. This approach enables us to use unsupervised or semi-supervised learning to detect zero-day attacks. The main flaw with this approach is that it might mistakenly report typical behavioral fluctuations as abnormalities, which increases the false positive rate.

Few writers use a standard called the anomalous score while implementing these systems. It presents the abnormality of a traffic sample based on NIDS[2] evaluation. A deep learning[3] model's computed loss function can be used to get to this number. For determining the anomaly score of the evaluated sample, we take the DNN's[9] output and compute its Euclidean distance to a fixed point, or take the anomaly score as a linear combination of discriminating loss and reconstruction of a bidirectional GAN[8]. A threshold that represents the upper bound for accepting anomalous activity is usually calculated by IDS that relies on anomaly score. Each time it exceeds the threshold, the respective traffic sample is classified as malicious. A number of methods, including mean, standard deviation and exhaustive search, can be used to analyse its value. The mean and standard deviation of the reconstruction errors for general occurrences determines the tolerance threshold. Fuzzy logic is used to define the anomaly score. A level of anomaly is determined using a Gaussian membership function. Attack validation data is utilized to determine the decision threshold by trial and error.

One of the major issues in developing an IDS for SDN[11] is to not overload the controller of the network. During data collection, making use of umpteen features for intrusion detection might cause an increased resource consumption and congestion on the southbound channel. When deployed on large networks, higher memory use and longer processing times might result in crashes, causing overhead, and controller bottlenecks. One of the other widely discussed issues is the lack of datasets that are exclusively SDN[11] and the possibility that they may not entirely capture the wide range of security concerns that exist in the real world. Ultimately deep learning[3] models can manage huge traffic volumes, a very important skill for the always increasing speeds of modern SDN[11] networks as a result of which, they are now the best option for developing intrusion detection systems.

Pattern extraction, classification, regression, synthetic data generation, and data distribution learning are just some of the computational issues in intrusion detection that are resolved by their ability to automatically capture complex structures in the data. The danger of controller overload is frequently lowered by feature selection and optimization without compromising the model's detection capabilities. The phases usually involved in developing DL-based NIDS, such as data collection and preprocessing, deep learning[3] modeling, hyperparameter tweaking, and performance evaluation, are depicted in Fig. 2.

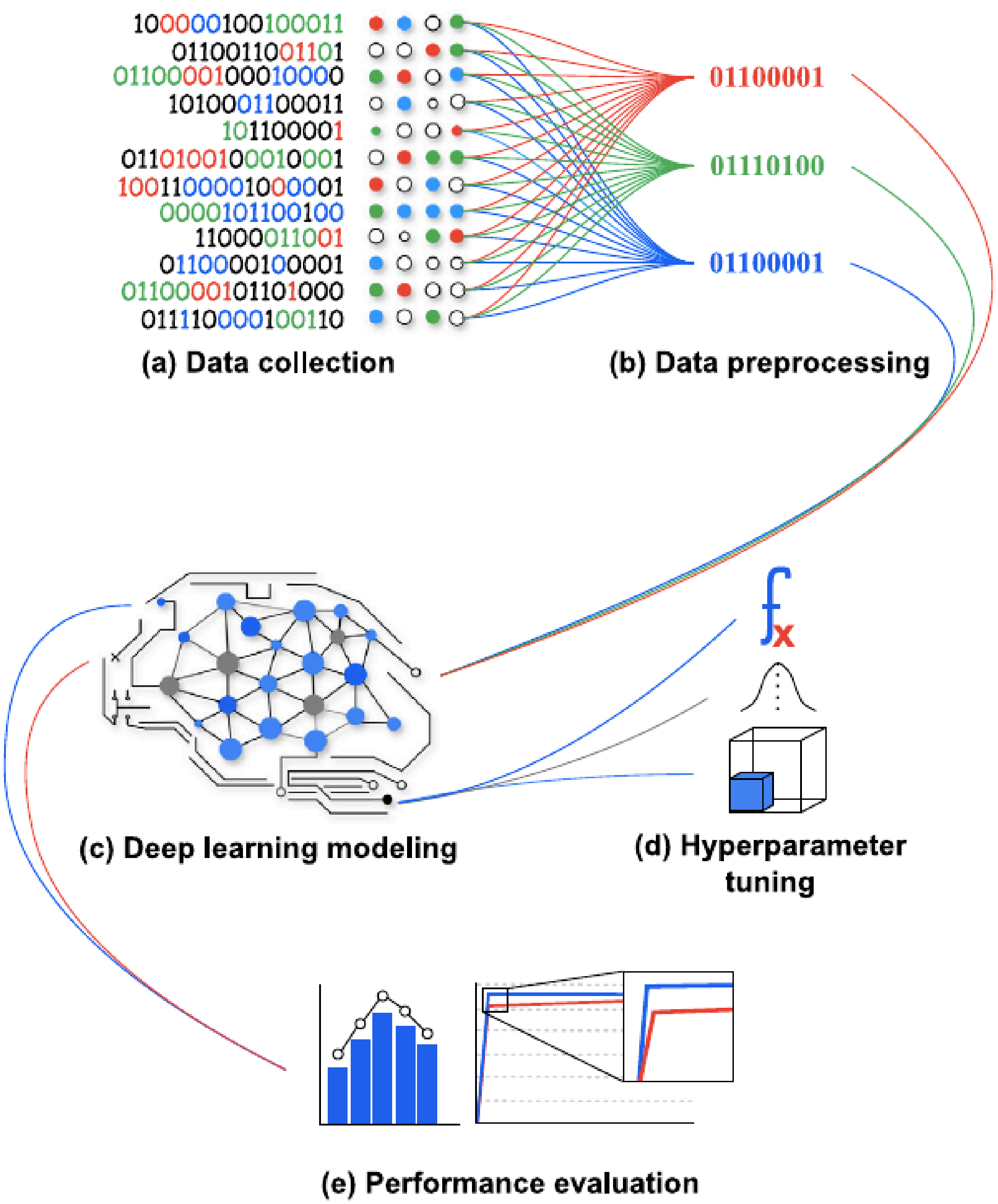

Fig 2. Deep learning-based NIDS development common framework.

RESULTS

7.1 Newborn Model:

Represents the initial state of the LSTM[4] before any training. At this stage, the model parameters are randomly initialized. The newborn model is like a blank state, unaware of the underlying patterns. So its predictions are likely to be arbitrary and inaccurate. This serves as the baseline, to indicate improvement through training.

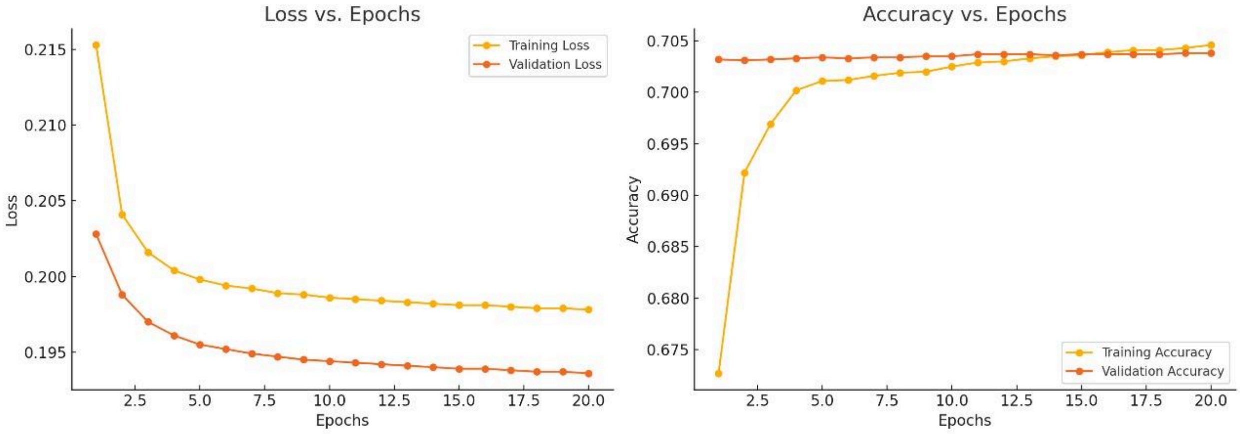

Fig 3. Results of the Newborn Model.

Detection Rate (Accuracy): 0.70385

Precision, Recall, F-score, Support: (0.30190630093185483, 0.3134835455761954, 0.281898991388258, None)

7.2 Dumb Model:

This time the model has been trained but fails to find patterns in data effectively. It might suffer from insufficient training, or oversimplification.

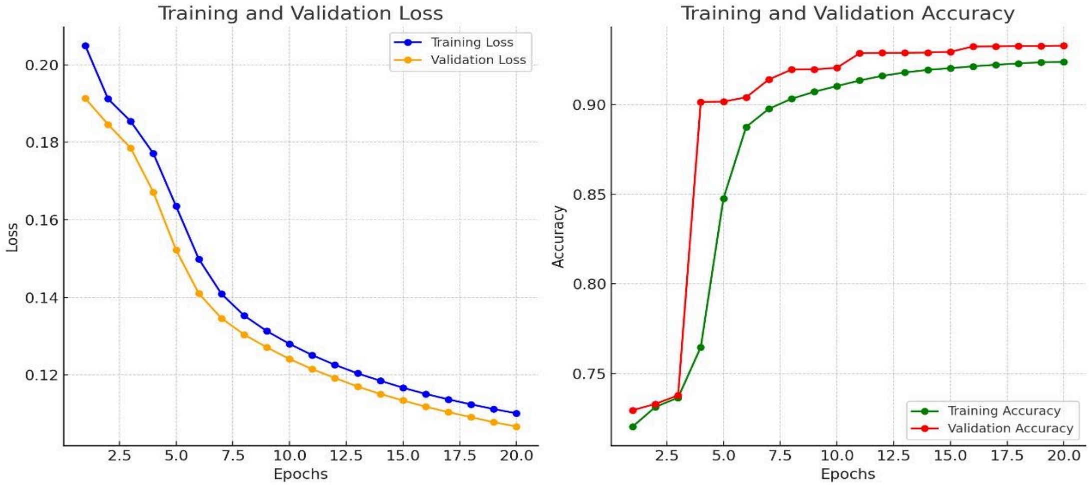

Fig 4. Results of the Dumb Model.

Detection Rate (Accuracy): 0.932925

Precision, Recall, F-score, Support: (0.3660848702580056, 0.38135916568518047, 0.3720734689192343, None)

7.3 Smart Model:

This time the model has been well-trained and optimized for the LSTM[4]. After thorough training, the model has acquired the ability to discern intricate patterns in the dataset. The smart model serves as the desired outcome.

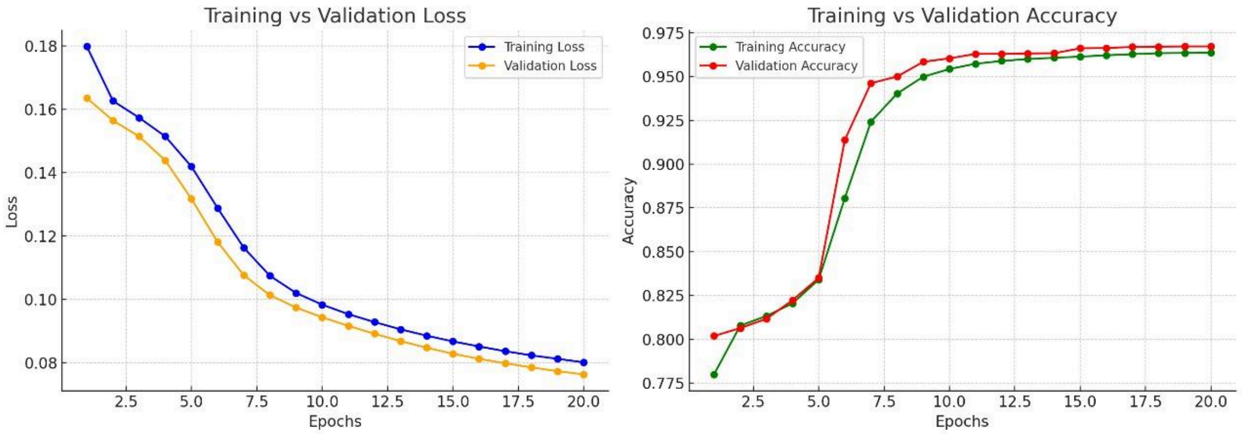

Fig 5. Results of the Smart Model.

Detection Rate (Accuracy): 0.96725

Precision, Recall, F-score, Support: (0.3828863120217367, 0.3915915193700597, 0.38692462952999385, None)

REFERENCES

1. da Silva Ruffo, V. G., Lent, D. M. B., Komarchesqui, M., Schiavon, V. F., de Assis, M. V. O., Carvalho, L. F., & Proença Jr, M. L. (2024). Anomaly and intrusion detection using deep learning for software-defined networks: A survey. Expert Systems with Applications, 124982.

2. Friha, O., Ferrag, M. A., Benbouzid, M., Berghout, T., Kantarci, B., & Choo, K. K. R. (2023). 2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT. Computers & Security, 127, 103097.

3. Hairab, B. I., Elsayed, M. S., Jurcut, A. D., & Azer, M. A. (2022). Anomaly detection based on CNN and regularization techniques against zero-day attacks in IoT networks. IEEE Access, 10, 98427-98440.

4. Tayfour, O. E., Mubarakali, A., Tayfour, A. E., Marsono, M. N., Hassan, E., & Abdelrahman, A. M. (2023). Adapting deep learning-LSTM method using optimized dataset in SDN controller for secure IoT. Soft Computing, 1-9.

5. Assis, M. V., Carvalho, L. F., Lloret, J., & Proença Jr, M. L. (2021). A GRU deep learning system against attacks in software defined networks. Journal of Network and Computer Applications, 177, 102942.

6. Fox, G. T., & Boppana, R. V. (2023). On early detection of anomalous network flows. IEEE Access, 11, 68588-68603.

7. Fouladi, R. F., Ermiş, O., & Anarim, E. (2022). A DDoS attack detection and countermeasure scheme based on DWT and auto-encoder neural network for SDN. Computer Networks, 214, 109140.

8. Shu, J., Zhou, L., Zhang, W., Du, X., & Guizani, M. (2020). Collaborative intrusion detection for VANETs: A deep learning-based distributed SDN approach. IEEE Transactions on Intelligent Transportation Systems, 22(7), 4519-4530.

9. Cil, A. E., Yildiz, K., & Buldu, A. (2021). Detection of DDoS attacks with feed forward based deep neural network model. Expert Systems with Applications, 169, 114520.

10. Nguyen, X. H., & Le, K. H. (2023). Robust detection of unknown DoS/DDoS attacks in IoT networks using a hybrid learning model. Internet of Things, 23, 100851.

11. Kumar, P., Kumar, R., Aljuhani, A., Javeed, D., Jolfaei, A., & Islam, A. N. (2023). Digital twin-driven SDN for smart grid: A deep learning integrated blockchain for cybersecurity. Solar Energy, 263, 111921.