IJARIIT
INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 11, Issue 4 - V11I4-1201)
Available online at: https://www.ijariit.com

A Network Security Monitoring System using Deep Learning

Pramodh Puthota
pramodhputhota@gmail.com
NRI Institute of Technology, Andhra Pradesh

MR. G.Sivannarayana
garikipati101@gmail.com
NRI Institute of Technology, Andhra Pradesh

Pasupuleti Bhavana Pradeepa Rani
pasupuletibhavana24@gmail.com
NRI Institute of Technology, Andhra Pradesh

Siriki Sravya
sravyasiriki2003@gmail.com
NRI Institute of Technology, Andhra Pradesh

Vaddeswarapu Rahul
vrahul200323@gmail.com
NRI Institute of Technology, Andhra Pradesh

ABSTRACT

In an era of evolving and increasingly complex cyber threats, the importance of robust network security is paramount. This paper presents a novel method of strengthening network defenses by building a highly flexible and durable Network Security Monitoring System (NSMS). By utilizing deep learning, more especially self-taught learning (STL), we set out to reinvent network security. In this study, we apply STL to the well-known NSL-KDD dataset, which is a commonly used network security monitoring system benchmark. We thoroughly analyze our NSMS solution's performance utilizing a range of important metrics, such as accuracy, precision, recall, and F-measure, to determine its overall effectiveness. Impressively, this method produced a 92.84% accuracy on the training set. As we use both the training and testing datasets in our work, our research expands on this basis and provides a distinct advantage for comparison, allowing a straight comparison to this earlier work. This study's main importance comes from its ability to prevent intentional attacks and to proactively identify unanticipated and unforeseeable security breaches. This research represents a milestone in the development of NSMS technology in the dynamic cybersecurity landscape, enabling enterprises to strengthen their security posture and protect their assets in a world that is becoming more interconnected.

Keywords—*Intrusion Detection, Network securing, Strengthening Network, Security Monitoring System.*

I. INTRODUCTION

Network system administrators can detect a multitude of security breaches in an organization's network architecture proactively by using Network Security Monitoring Systems (NSMSs), which are an essential tool in their toolbox. These watchful systems carefully monitor every network traffic entering and leaving, recording, examining, and sounding the alert as needed. Based on how they detect intrusions, NSMSs are divided into two main categories: i) signature-based NSMS (SNSMS) and ii) anomaly detection-based NSMS(ADNSMS) [1].

The rules governing known attacks come pre-installed in the world of SNSMS; a good example of this is Snort. SNSMS performs a complex pattern matching procedure, comparing the observed traffic against these predetermined rules, to identify potential network intrusions. SNSMS is highly effective in identifying known threats, providing a high detection accuracy and low false alarm rate. Nevertheless, the drawback is that performance can deteriorate in the event of unfamiliar or new attacks due to the rigidity of using only pre-installed rules. On the other hand, ADNSMS employs a more flexible strategy, identifying any changes from the typical traffic patterns as intrusions and reporting them as such. With a remarkable degree of adaptability that SNSMS might not have, ADNSMS proves to be a powerful tool in identifying unknown and novel threats. The research community has embraced and enthusiastic about ADNSMS's theoretical potential for identifying novel attacks, even with the notable trade-off of producing a significant number of false positives.

There are two major obstacles to overcome in the development of a flexible and effective Network Security Monitoring System (NSMS). First off, choosing the best features for anomaly detection is a dynamic and challenging process due to the constantly changing nature of cyber threats. What is effective against one type of attack might quickly become insufficient against new threat scenarios. Leveraging the power of AI and machine learning, which can adapt to changing threats, is essential to overcoming this challenge. Secondly, the development of NSMS is hampered by the lack of labeled datasets from actual network environments. Building these kinds of datasets from actual network traffic traces takes a lot of effort and money, which is made more difficult by network administrators' unwillingness to divulge information to protect user privacy and the secrecy of internal network architecture [2].

The creation of artificial datasets that accurately mimic network activity, the use of anonymization and privacy-preserving techniques to enable data sharing, and the promotion of cooperative efforts between institutions, researchers, and government agencies to produce shared, labeled datasets while adhering to strict privacy regulations are all potential solutions. Maintaining privacy while bolstering cybersecurity requires careful consideration, especially as new developments in technology and research continue to address these crucial issues in the field of cybersecurity.

In the field of network security, Network Security Monitoring Systems (NSMSs) are essential because they act as the first line of defense against online attacks. NSMSs use a wide range of machine learning techniques, such as Random Forests, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Naive-Bayesian (NB), Self-Organized Maps (SOM), and more, to distinguish between normal network traffic and potentially malicious or abnormal activity. By acting as classifiers, these methods decide in real time whether to flag or reduce questionable network activity.

The phase of NSMS development has seen a radical transformation in recent years with the incorporation of deep learning methodologies. Large amounts of unlabeled network traffic data can be used to create complex patterns and representations that are easily learned by deep learning models, especially deep neural networks. NSMSs are able to comprehend and adjust to changing threats thanks to this feature learning process. Deep learning performs best when there is a lack of labeled data, which is frequently the case in network security. By applying the knowledge gathered from a large pool of unlabeled data to a smaller dataset that has labels, it closes the gap and makes supervised classification easier [3].

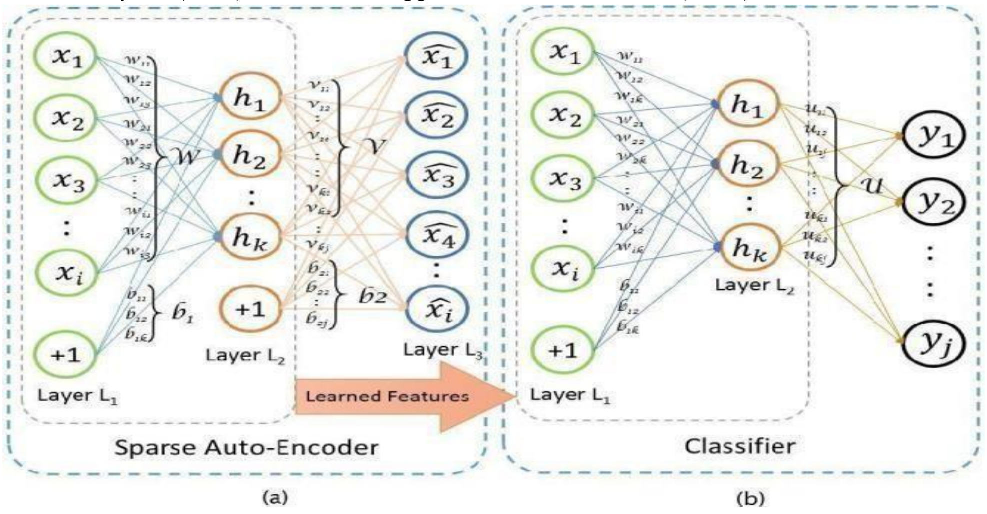
Our methodology is based on the use of self-taught learning, a potent deep learning technique that makes use of sparse auto-encoders and soft-max regression. Self-taught learning performs exceptionally well at automatically deriving valuable features from unprocessed network data, enabling the NSMS to adjust to novel and changing threats. Through extensive experimentation on the NSL-KDD intrusion dataset, we hope to show the viability and effectiveness of this approach. This dataset is a refined version of the KDD Cup 99 benchmark that provides a wide range of detailed network traffic records, which makes it a perfect tool for performance evaluation of NSMS.

To accomplish this, our paper is structured into four sections. In Section 2, we review a few closely related works. Section 3 gives a summary of the NSL-KDD dataset and self-taught learning. We discuss our results and comparative analysis in Section 4. We conclude our work and offer suggestions for more research in Section 5.

II. RELATED WORK

This section features several recent accomplishments in this field. Please take note that the NSL-KDD dataset is the only one used for performance benchmarking in the works we discuss; henceforth, NSL- KDD will be used to refer to any dataset mentioned. This enables more accurate comparisons between our work and the works of other authors. Most of the work uses training data for both training and testing, which is another disadvantage. Finally, we discuss some of the deep learning-based techniques that have been used so far in this field.
An ANN with improved resilient backpropagation was employed in one of the first studies for the design of such an IDS that could be found in the literature [6]. In this work, training (70%), validation (15%), and testing (15%) were conducted using only the training dataset. Performance decreased when testing with unlabeled data, as would be expected. The J48 decision tree classifier was employed in a more recent study, and testing was carried out once more using only the training dataset and 10-fold cross validation [7]. Out of the 41 features in total, only 22 were used in this work. A related study evaluated a number of well-known supervised tree-based classifiers in detail with the goal of determining how well they performed in terms of accuracy and false alarm rate [8]. The results of extensive testing and analysis showed that the Random Tree model consistently performed better than the others.

There have been several proposed two-level classification techniques in the field of intrusion detection. One noteworthy study used a 10- fold cross-validation strategy for testing. Using Discriminative Multinomial Naive Bayes (DMNB) as the base classifier, Nominal to Binary supervised filtering was applied at the second level [9]. END (Ensembles of Balanced Nested Dichotomies) was introduced at the first level and Random Forest at the second level in a subsequent study that built upon this foundation [10]. As expected, this improvement produced a significant decrease in the false positive rate and an increase in the detection rate. The authors used a different two-level implementation in which they reduced the size of the feature set using Principal Component Analysis (PCA) and used Support Vector Machines (SVM) with Radial Basis Function for final



classification. Surprisingly, this method produced a high detection accuracy with just the training dataset and all 41 features. Reducing the feature set to 23 resulted in even better detection accuracy for some attack classes, but at the cost of decreased overall performance [11]. The reported accuracy showed significant improvements because of these feature selection strategies and methodological improvements. The accuracy of the system was significantly increased by the careful integration of methods like Principal Component Analysis (PCA) for feature reduction, Random Forest, END (Ensembles of Balanced Nested Dichotomies), Nominal to Binary supervised filtering, and Discriminative Multinomial Naive Bayes.

Regarding the second area of analysis, the performance of different intrusion detection techniques was assessed using both training and testing datasets. In one notable attempt in this category, fuzzy classification was combined with a genetic algorithm, leading to an impressive achievement of over 80% detection accuracy and a low false positive rate [13]. It is noteworthy, though, that performance obtained with just the training data was noticeably worse in another significant study that considered both the training and testing data. This study investigated intrusion detection using unsupervised clustering algorithms [14]. Like this, another implementation made use of the k-point algorithm and produced a slight but noticeable improvement in false positive rates and detection accuracy by utilizing both the training and test datasets [15]. Interestingly, the less well-known OPF (Optimum-Path Forest) method—which uses graph partitioning for feature classification—showed a remarkably high detection accuracy when applied to Intrusion Detection Systems (IDS) [16]. Surprisingly, OPF performed better than the popular SVM-RBF approach, attaining comparable outcomes in a third of the time.

A deep learning approach was investigated in a prior study [5], using a Deep Belief Network (DBN) as the feature selector and a Support Vector Machine (SVM) as the classifier. Impressively, this method produced a 92.84% accuracy on the training set. As we use both the training and testing datasets in our work, our research expands on this basis and provides a distinct advantage for comparison, allowing a straight comparison to this earlier work. It is noteworthy, nevertheless, that a different study [2] also used a semi-supervised learning strategy that is somewhat like our approach. The KDD Cup 99 dataset and actual network traces were both used by the study's authors to evaluate their methodology. Notably, they used actual network traces to train their model. On the other hand, our method makes use of the NSL-KDD dataset and seeks to evaluate the suitability of deep learning for the domain of Network Security Monitoring Systems (NSMS). It is interesting to note that a recent study [17] also presented a sparse auto-encoder- based deep learning technique for network traffic identification. It is noteworthy, nevertheless, that this work concentrated on the TCP identification of unknown protocols rather than the area of network security monitoring System

III. LITERATURE REVIEW

Certainly, let us organize the Network Security Monitoring Systems (NSMS) literature review into paragraphs:
A vital part of contemporary cybersecurity are Network Security Monitoring Systems (NSMS), which are made to recognize and react to attacks, unauthorized access, and unusual network activity. The two primary types of these systems are broadly classified as anomaly- based and signature-based. Anomaly-based systems create a baseline of typical network behavior and alert users for any deviations, while signature-based systems rely on predefined attack signatures to detect known intrusion patterns in the complex world of Network Security Monitoring Systems (NSMS). The ongoing issue in this area is the high frequency of false positives produced by NSMS, which overwhelm security personnel with a deluge of alerts. Acknowledging this problem, scientists have worked nonstop to improve NSMS accuracy, aiming to lower false positives and increase overall effectiveness.

The use of machine learning, particularly deep learning techniques, has become increasingly popular in the dynamic field of Network Security Monitoring Systems (NSMS). Their ability to identify and adjust to new attack patterns has great potential to improve the accuracy of intrusion detection. This adaptation makes it possible to process and analyze large datasets more effectively, which is very helpful in identifying subtle attack patterns that may evade traditional methods. The integration of deep learning, machine learning, and big data analytics represents a significant advancement in Network Security Monitoring Systems (NSMS), enabling these systems to traverse the intricate landscape of cybersecurity more accurately and adaptively.

The emergence of IoT and cloud computing has presented new difficulties for NSMS. The goal of research has shifted to creating NSMS that can watch over and safeguard cloud-based networks. Constrained resources and a variety of communication protocols in IoT environments make it necessary to develop lightweight NSMS solutions that can meet the requirements of IoT devices. This entails addressing IoT-specific privacy issues, data integrity issues, and device vulnerabilities.

The use of behavioral analysis in NSMS research is a recent development. This is a subset of anomaly-based detection that is centered on ongoing observation and learning from network and user behavior. Cutting-edge behavioral analysis methods seek to identify complex attacks that might not have signatures and adjust to shifting network dynamics. Moreover, hybrid NSMS strategies have been investigated, fusing anomaly-based and signature-based techniques to maximize each strategy's advantages and increase detection accuracy.

Performance and scalability are critical components of NSMS development as networks get bigger and more complex. Large data volumes must be processed by these systems in an efficient manner with minimal latency, particularly in high-traffic environments. Furthermore, by keeping up with the most recent threat intelligence, the integration of threat intelligence feeds into NSMS has become standard procedure, improving their capacity to identify and address emerging threats.

Certainly, let us elaborate on the Network Security Monitoring Systems (NSMS) literature review:

Network Security Monitoring Systems (NSMS) are essential protectorsin the vast field of network security. Their main objective? to monitor network activity closely and act quickly to spot any suspicious or malicious activity that might jeopardize a network's integrity or security. The silent observers, or Network Security Monitoring Systems, function in two elegant ways: as rule-bound, signature-based systems or as flexible, anomaly-based counterparts.

With their arsenal of pre-established patterns or signatures of known attacks, signature- based NSMS are akin to seasoned detectives equipped with a list of known criminal profiles. Conversely, anomaly- based NSMS take a more dynamic stance, establishing a standard for typical networkbehavior.

IV. METHODOLOGY

4.1 Self-Taught Learning

Two stages make up the deep learning method known as self-taught learning (STL) for classification. Initially, to acquire an accurate feature representation, a significant quantity of unlabeled data, or xu, is utilized. This process is known as unsupervised feature learning (UFL). Using labelled data (xl) and the learned representation, the classification task is carried out in the second stage. There must be a relationship between the labelled and unlabeled data even though they may originate from different distributions. The STL architecture diagram is displayed in Figure 1. Various techniques are employed for UFL, including Gaussian Mixtures Sparse Auto-Encoder, K-Means Clustering, and Restricted Boltzmann Machine (RBM) [19]. We employ sparse auto-encoder based feature learning in our work because of its excellent performance and simplicity of implementation [4]. A sparse auto-encoder is a neural network composed of input, hidden, and output layers. The input and output layers contain N nodes each, while the hidden layer contains K nodes. The output layer's target values, as shown in Figure 1(a), are set to the input values, or φxi = xi. In the process of learning the identity function approximation—that is, output xˆ simi lar to x—the sparse auto-encoder network employs a backpropagation algorithm to determine the ideal weight matrix values, together with the bias vectors, b1 ∈ KX1 and b2 ∈ ~NX1, W ∈ KXN and V
∈ RNXK [18]. g(z) = 1, the sigmoid function

The cost function that must be minimized in a sparse auto-encoder through backpropagation is shown in Equation 2. The first term is the average of the sum-of-square errors term for all m input data. The second term is a weight decay term with A as the weight decay parameter, preventing over-fitting during training. The hidden layer is constrained to maintain low average activation values by the final term in the formula, referred to as the sparsity penalty term, which is expressed as the Kullback-Leibler (KL) divergence found in Equation 3.

The sparsity constraint parameter p, which varies from 0 to 1, governs the sparsity penalty term. At p = \pj, where ˆpj is the average activation value of hidden unit j across all training inputs, x, the KL(p~\pj) approaches its minimum value. Once the ideal values for W and b1 have been determined by applying the sparse auto-encoder to the unlabeled data (xu), we evaluate the feature representation a = hW,b1(xl) for the labelled data (xl, y). We combine the labels vector, y, with this new feature representation, a, for the classification task of the second stage. To perform the classification task, we use soft-max regression, as shown in Figure 1(b) [21].

4.2 NSL-KDD Dataset

In the complex world of intrusion detection, where network security and data science converge, the NSL-KDD dataset acts as a guide for us. This is an enhanced version of the well-known KDD Cup 99 dataset, which captures the essence of network traffic that was painstakingly captured during the 1998 DARPA IDS test program [22]. This digital mosaic includes the regular ups and downs of network activity as well as the covert dance of different attack kinds, ranging from the infamous Denial of Service (DoS) to probing techniques. The neural networks learn from the complex nuances of this network symphony over the course of our seven- week training voyage [20]. Two weeks of unfiltered traffic are set aside for testing as the ultimate test, adding a dash of unpredictability along with a multitude of attacks that aren't in the training set. It is a realistic journey through intrusion detection where the neural guardians must separate the known from the unknown. Our dataset, which contains millions of TCP/IP connection records, opens a wide canvas on which the skill of recognizing and classifying novel and evolving attacks can be showcased.

The KDD Cup dataset is the industry standard for NSMS evaluations. Nevertheless, a serious flaw mars its glory: duplicate records are common in both test and training sets. Imagine that there is an astounding 78% redundancy in the training data and 75% in the test data, which leads to a maze of identical pathways for algorithms to learn [20]. Because of this redundancy, algorithms are biased towards the well-known territory of frequently occurring attack records, which tips the scales. What is the result? Erroneous categorization of infrequent but potentially dangerous occurrences gets lost in the din. The dataset has an amazing minimum accuracy of 98% in training and 86% in test set, even with basic machine learning algorithms. This presents a challenge that makes the search for meaningful comparisons more difficult. It's like comparing various IDSs and learning algorithms in a chaotic landscape where most enemies wear the same disguise.

By using a two-pronged approach to mitigate its limitations, NSL- KDD emerges as a strategic enhancement to the KDD Cup dataset. First off, cleaner, more reliable data is created when duplicate records are removed from both the training and test sets. Second, each record in the dataset is carefully categorized according to its degree of difficulty, which is determined by how well different learning algorithms classify the data. A random sampling technique that is inversely proportional to the fraction of records within each difficulty level is made possible by this stratification, which also serves to refine the dataset. Because the original KDD Cup dataset was thoughtfully processed in multiple steps, the resulting NSL-KDD dataset presents a realistic and well-balanced collection of records, ready to effectively train a spectrum of learning algorithms.

The NSL-KDD dataset provides an extensive view, with each record containing 41 features and classified as either normal or indicative of a particular kind of attack. These features are a complex combination of traffic features (like time or connection count) collected within predefined window intervals, content features extracted from application layer data, and core features directly from TCP/IP connections. With 34 continuous, 3 nominal, and 34 binary features, the dataset has a wide range of features. There are 23 traffic classes in the training set, consisting of 22 attack classes and 1 normal class. With 21 attacks incorporated into the 38 traffic classes, the test data further increases this complexity, offering a solid and complex basis for developing and accessing machine learning models.

The NSL-KDD dataset is a widely used and well-known resource in the field of cybersecurity and network security monitoring System. It's an improved version of the KDD Cup 1999 dataset, which was originally created for the DARPA Intrusion Detection Evaluation Program. The principal objective behind the creation of the NSL- KDD dataset was to address certain deficiencies of its forerunner, such as data duplication and erroneous distribution of data. The NSL- KDD dataset was essentially selected to offer a more complete and representative set of information for the evaluation and creation of Network Security Monitoring Systems (NSMS). This collection contains a great deal of information, including network traffic data and labels that indicate intrusion occurrences. It combines a wide variety of legitimate and malicious network traffic. The data itself consists of various network connection features, including protocol types, services, duration of connections, source and destination IP addresses, and more. The NSL-KDD dataset is notable for its ability to classify attacks into four main categories: "Normal," "DoS" (Denial of Service), "Probe," and "R2L" (Remote to Local. Every class denotes a unique kind of network intrusion, enabling a thorough assessment of intrusion detection systems in a range of threat scenarios. The original KDD Cup 1999 dataset had several flaws that the NSL-KDD dataset helped to fix by providing a more accurate and balanced depiction of network traffic. To create and assess intrusion detection algorithms, researchers primarily split this dataset into training and testing subsets. Because of this, it is now a vital tool for researchers and cybersecurity professionals who want to evaluate, contrast, and improve the effectiveness of various intrusion detection systems.

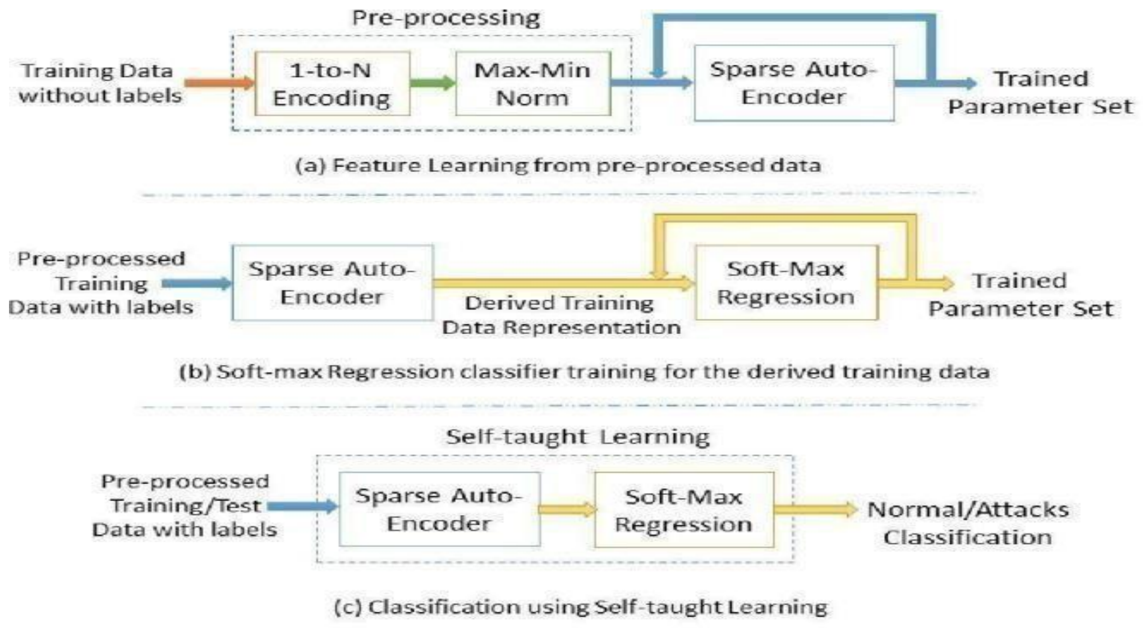

Pre-processing
Training Data without labels
1-to-N Encoding
Max-Min Norm
Sparse Auto-Encoder
Trained Parameter Set
(a) Feature Learning from pre-processed data
Pre-processed Training Data with labels
Sparse Auto-Encoder
Derived Training Data Representation
Soft-Max Regression
Trained Parameter Set
(b) Soft-max Regression classifier training for the derived training data
Self-taught Learning
Pre-processed Training/Test Data with labels
Sparse Auto-Encoder
Soft-Max Regression
Normal/Attacks Classification
(c) Classification using Self-taught Learning

Figure 2: The different stages of our NSMS implementation

V. RESULTS AND DISCUSSION

The assessment environment for Network Security Monitoring Systems (NSMS) is characterized by two main approaches. The current method uses n-fold cross-validation, which produces significantly higher accuracy and lower false-alarm rates, and carefully divides training data into test, training, and cross-validation sets. On the other hand, an alternative methodology separates the training and test data for different purposes, taking a different route. Because of the various settings in which the data was collected, this second approach provides a more nuanced assessment even though it achieves a lower accuracy than its counterpart. In our work, we diligently highlight the results of this latter strategy, emphasizing accuracy in NSMS assessment. Concurrently, we exhibit the outcomes of the initial approach, offering an all-encompassing viewpoint on the functionality of the system. An insightful summary of our NSMS implementation is provided before diving into the results to help you understand the resulting results more deeply.

5.1 NSMS Implementation

In our self-taught learning approach, the careful preparation of the dataset is central to building on the diverse attributes and values that are inherent in the data. In order to improve compatibility, we use the 1-to-n encoding technique to convert nominal attributes into discrete counterparts. Simplifying even more, we find and remove an attribute that is always set to 0. This attribute is present in both training and test records. After completing these preprocessing stages, the dataset is refined and has 121 attributes in total. The sigmoid function is used in the feature learning phase, as shown in Figure 1(a), to compute values in the output layer that smoothly range from 0 to 1. Notably, the input layer values are shaped within the 0 to 1 spectrum by the normalization effect caused by the alignment of the output layer values with their corresponding input layer values. This deliberate normalization establishes the groundwork for the later stages of our self-directed learning project. Training the deep learning model involves using the preprocessed data to teach the model to distinguish between benign and malicious traffic patterns. This process may involve hyperparameter tuning, cross-validation, and fine- tuning to achieve the best performance.

Validation and testing follow model training to ensure its effectiveness and generalization. The NSMS must be capable of accurately identifying network intrusions while minimizing false positives. Fine-tuning and retraining may be necessary based on the performance evaluation results. Finally, the deployed NSMS continually monitors network traffic, classifying it in real-time. When malicious activity is detected, appropriate actions are taken, such as alerting administrators or implementing automatic mitigation measures.

Validation and testing follow model training to ensure its effectiveness and generalization. The NSMS must be capable of accurately identifying network intrusions while minimizing false positives. Fine-tuning and retraining may be necessary based on the performance evaluation results. Finally, the deployed NSMS continually monitors network traffic, classifying it in real-time. When malicious activity is detected, appropriate actions are taken, such as alerting administrators or implementing automatic mitigation measures.

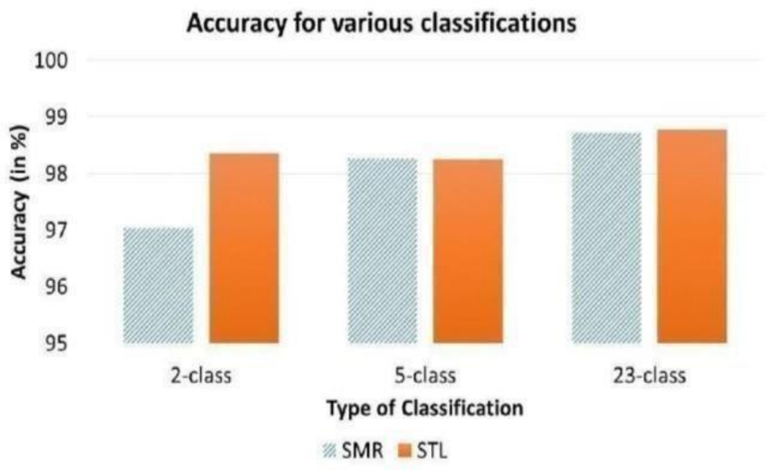


Figure 3: Classification accuracy using self-taught learning (STL) and soft-max regression (SMR) for 2-Class, 5-Class, and 23-Class when applied on training data

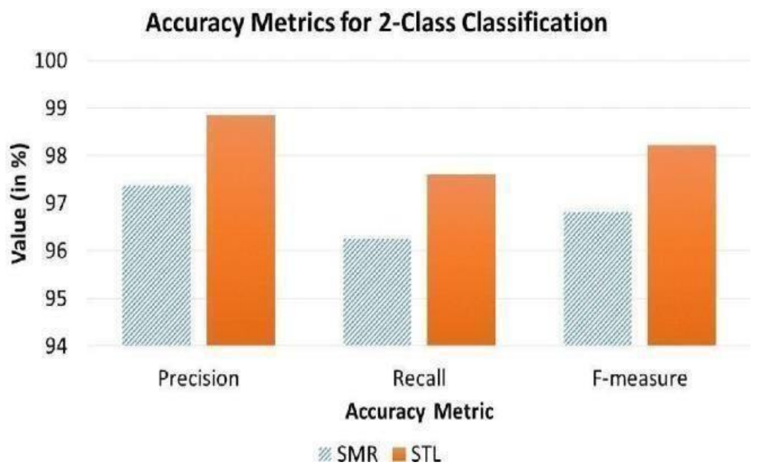


Figure 4: F-Measure, Precision, and Recall values for 2-Class using soft-maX regression (SMR)and self-taught learning (STL) applied to training data

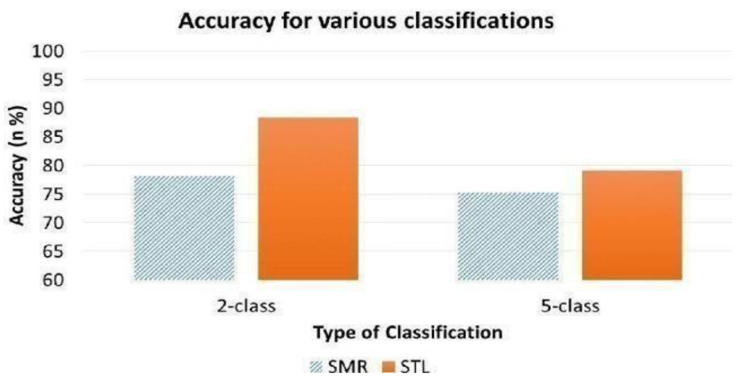
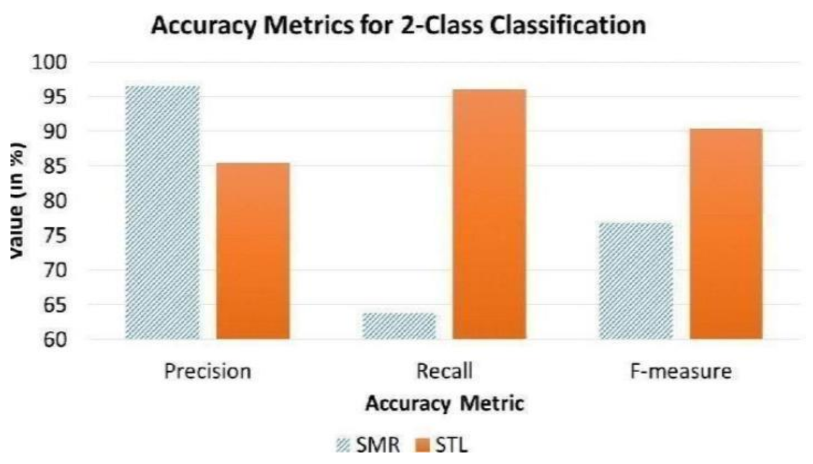


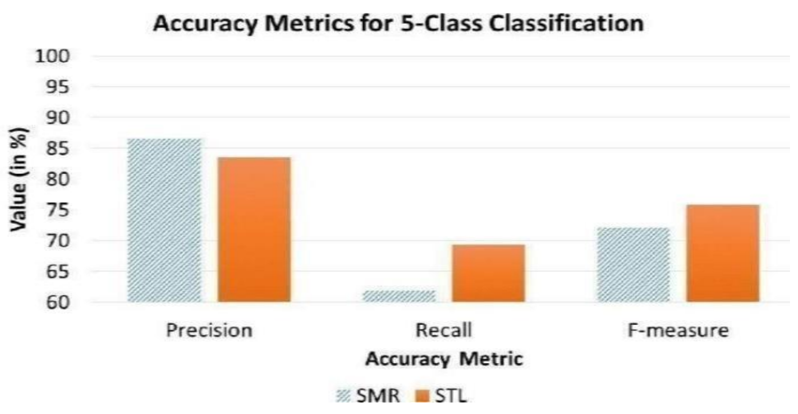
Figure 5: When applied to test data, self-taught learning (STL) and soft-max regression (SMR) for 2- and 5-class classification accuracy

There are a few essential steps involved in putting into practice a deep learning strategy for Network Security Monitoring Systems (NSMS). The first step is data collection, which involves massing a sizable dataset of network traffic that includes both malicious and legitimate traffic.

Accurately labelling the data is essential for differentiating between safe and dangerous network flows. Preprocessing data comes next. This phase includes gathering and analyzing the unprocessed network data, which is frequently available as network flow records (NetFlow data) or packet capture files (PCAPs). It is necessary to extract pertinent features like protocols, timestamps, packet sizes, ports as well as source and destination IP addresses. To make the data uniform and suitable for training deep learning models, scaling and normalization are also applied.

Next, the deep learning model is the brains behind the NSMS. Popular options for this task are Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). While RNNs are superior at handling sequential data, which is essential for network traffic analysis, CNNs are good at identifying spatial patterns in data, which makes them appropriate for identifying irregularities in network traffic.





Figures 6 and 7 show the F-Measure, Precision, and Recall values for 5-class using soft-max regression (SMR) and self-taught learning (STL) on test data.

To achieve an efficient Network Security Monitoring System (NSMS), we must optimize our attribute list, which requires careful max-min normalization to guarantee a common scale for the various attributes. With NSL-KDD training data, the path toward self-taught learning is revealed by purposefully removing labels to promote feature learning. In this case, the sparse auto-encoder is essential because it explores the nuances of the data to extract insightful representations.

The second phase is identified by a seamless transition, in which soft- max regression is used to classify the training data using the recently

obtained feature representation as a reference point. Specifically, our implementation strategy is unique in that it uses labeled and unlabeled NSL-KDD training data to create a seamless story that connects classifier training and feature learning. Figure 2 skillfully illustrates the complex dance of these processes, offering a visual road map for the subtle actions made in developing a reliable and knowledgeable intrusion detection system.

5.2 Accuracy Metrics

The following metrics are used to assess how well self-taught learning performs:

i. Accuracy: Measured as the proportion of accurately classified records to all records.
ii. Precision (P): Defined as the percentage ratio of the total number of classified true positive (TP) records divided by the total number of classified false positive (FP) records.

a. TP

2. P = (TP + FP) × 100% (4)

iii. Recall (R): represented as the fraction of classified false negative (FN) and true positive records divided by the percentage of true positive records.

TP

R = (TP + FN) × 100% (5)

VI. CONCLUSION AND FURUTE WORK

Our suggested method for creating a versatile and efficient NSMS is based on deep learning. An NSMS based on SoftMax regression and sparse auto-encoder was put into practice. To assess anomaly detection accuracy, we used the NSL-KDD benchmark network intrusion dataset. When tested with test data, our Network Security Monitoring System (NSMS) performed remarkably well, surpassingits peers in the areas of normal and anomaly detection. Its effectiveness was further enhanced by the deliberate integration of classification strategies like NB-Tree, Random Tree, and J48, as well as the reinforcement of unsupervised feature learning via the Stacked Auto-Encoder, a powerful expansion of the sparse auto-encoder in deep belief nets. Notably, as previous research has shown, using these techniques directly on the dataset produced encouraging results [20]. Our goal is to build a real-time Network Security Monitoring System (NSMS) that is specifically designed for real-world network environments by utilizing deep learning techniques

REFERENCES

[1] Lirim Ashiku and Cihan Dagli explored the application of deep learning techniques to enhance the effectiveness of intrusion detection systems. Published in Procedia Computer Science. https://doi.org/10.1016/j.procs.2021.05.025

[2] Alazab, M., Layton, R., & Venkatraman, S. (2011). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19-31.

[3] García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network security monitoring System: Techniques, systems, and challenges. Computers & Security, 28(1-2), 18-28. https://doi.org/10.1016/j.jnca.2015.11.016

[4] Ahmad et al. (2020) ,conducted a comprehensive study titled "Network intrusion detection system: A systematic study of machine learning and deep learning approaches." This research systematically reviews the application of machine learning (ML) and deep learning (DL) techniques in Intrusion

[5] Detection Systems (IDS), highlighting their potential in enhancing detection accuracy and reducing false positives. https://doi.org/10.1002/ett.4150

[6] "A Deep Learning Approach for Network Intrusion Detection System," Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam (2016) explored the application of deep learning techniques to enhance the effectiveness of Network Intrusion Detection Systems (NIDS). Published in the EAI Endorsed Transactions on Security and Safety. https://doi.org/10.4108/eai.3-12-2015.2262516

[7] Cloud and NSMS:
Zhang, R., & Qian, H. (2018). A Survey of Cloud Intrusion Detection System: Research Challenges and Opportunities. Journal of Cloud Computing: Advances, Systems https://doi.org/10.1016/j.jnca.2016.10.015

[8] IoT and NSMS:
- Alaba, F. A., Othman, M., Hashem, I. A. T., Alotaibi, F., & Zolkipli, M. F. (2017). Internet of

[9] Things security: A survey. Journal of Network and Computer Applications, 88, 10-28.

[10] Su, M. Y., & Chen, K. T. (2017). A Survey of Intrusion Detection in Internet of Things. Journal of Network and Computer Applications, 84,25
37.https://doi.org/10.1016/j.jnca.2017.04.002

[11] Behavioral Analysis:
i. Zanero, S., & Hauser, R. (2005). Network anomaly detection: A machine learning

[12] perspective. In Recent Advances in Intrusion Detection (pp. 122-142). Springer.https://doi.org/10.1201/b15088

[13] Hybrid Approaches:
i. Hodge, J. J., & Austin, T. H. (2004). A survey of outlier detection methodologies. Artificial Intelligence Review, 22(2), 85-126.https://link.springer.com/article/10.1007/s10462-004-4304-y

[14] Scalabilityand Performance:
i. Casado, L. G., Esparza, S. G., & Casado, S. G. (2017). Scalable high-performance deep packet inspection on many-core architectures. In 2017 IEEE Symposium on Computers and Communications (ISCC) (pp. 1165-1170). IEEE. https://doi.org/10.1109/ISCC.2017.8024700

[15] Md. Asadul Hoque and Md. Moniruzzaman (2022) conducted an extensive review of deep learning methodologies applied within cybersecurity contexts. Published in Computers & Security.
https://doi.org/10.1016/j.cose.2022.102861