The International Journal of Business Management and Technology, Volume 9 Issue 2 February 2025
ISSN: 2581-3889

Research Article Open Access

Development of A Hybrid Network Intrusion Detection Using Machine and Deep Learning Algorithms

Engr. Udoawujo Bismarck Maduagwu [1], Prof Joseph.M. Mom [2], Engr. Dr. Ephraim Tersoo Iorkyase [3]

(udbismarck@gmail.com)[1]
(joe.mom@uam.edu.ng)[2]
(Ephraim.iorkyase@uam.edu.ng)[3]

ABSTRACT: *This paper proposes a hybrid approach to Intrusion Detection Systems (IDS) by combining machine learning algorithms with signature-based methods. The study explores how the integration of Long Short-Term Memory (LSTM), Support Vector Machine (SVM) and Artificial Neural Networks (ANN) with traditional signature-based detection can offer a more robust solution to cybersecurity challenges. The hybrid model addresses both known and unknown threats, leveraging the strengths of each technique. Through data collected from network traffic logs, this paper demonstrates how the hybrid IDS can enhance accuracy and adaptability in detecting malicious activity.*

Keywords: *Hybrid Intrusion Detection System, Machine Learning, Signature-based Detection, Support Vector Machine, Long Short-Term Memory (LSTM), Artificial Neural Networks, Cybersecurity.*

I. INTRODUCTION

Technological advancements have led to a rise in network intrusion and cybercrime, which pose serious threats to private companies and governments. Network intrusion involves unauthorized access or attacks on computer systems or networks, while cybercrime involves illicit conduct involving computer or internet use. Both threats can result in financial loss, data breaches, and physical harm. To mitigate these threats, precautions and strong security measures must be implemented, such as using strong passwords, regularly updating software, and exercising caution when sharing personal information online.

The growing concern about network intrusion has led to research on intrusion detection systems, which monitor network traffic and system activity for signs of unauthorized access, misuse, or malicious activity. There are two primary types of intrusion detection systems: signature-based and behavior-based. Signature-based systems look for specific patterns of known attacks, while behavior-based systems use machine learning techniques to assess network activities and detect unusual or suspicious behavior.

The increasing rate of cybercrime has led to the need for strong security measures like Intrusion Detection Systems (IDSs) to identify and respond to attacks in real time. IDSs can help identify malicious activities and prevent data breaches by monitoring network traffic, identifying anomalies and suspicious patterns, and alerting security teams to potential threats.

Intrusion detection systems (IDS) in Machine Learning (ML) systems are crucial for detecting and responding to threats in real time, identifying sophisticated attacks that may go unnoticed by traditional security tools, and improving the accuracy and reliability of ML models. This study proposes the application of machine learning algorithms like Multi-Layer Perceptron, Support Vector Machine, and the Long Short-Term Memory algorithm for detecting intrusion from a dataset sourced from the Communication Security Establishment and Canadian Institute for Cyber security.

IDS can be categorized into four types: Signature-based, Anomaly-based, machine learning-based, and deep learning-based IDS. Machine learning IDS uses learnable algorithms from data to identify patterns in network traffic, which could indicate a potential security threat. It is less prone to generate false positives compared to anomaly-based IDSs and can adapt to changing network traffic patterns and evolving threats. Long Short-Term Memory (LSTM) is a deep learning technique that operates on sequential data and can learn long-term dependencies in sequential data, making it suitable for tasks like language translation, speech recognition, and time series forecasting.

In summary, machine learning IDS offers a more accurate, adaptive, and effective approach to detecting security threats than other technology-based IDS. However, it requires significant amounts of data to be trained effectively and may demand additional computing power and resources. Therefore, the research will involve the administration of both machine and deep learning models, including a combination of signatures, anomaly, and behavior-based approaches, to develop an Intrusion detection system to identify and react to a diverse spectrum of cyber threats.

II. LITERATURE REVIEW

Intrusion Detection Systems (IDS) are security mechanisms that continuously analyze host and network traffic to identify illicit activities that may compromise system security. They are designed to monitor information sources like computers or networks for unlawful activities and can be implemented using various strategies and tactics. The proliferation of local networks and the Internet has led to a surge in intrusion incidents targeting computer systems. IDSs must deal with difficulties such as high traffic volume and unequal data distribution. Network Intrusion Detection Systems (NIDS) can be implemented using three detection techniques: signature-based detection and anomaly-based detection. Signature-based NIDS only detects known malicious threats, while anomaly-based NIDS automatically comprehends unknown and erratic attacks. Detecting covert attacks amidst numerous legitimate communication transactions poses a significant challenge in intrusion detection.

Machine learning (ML) is a field that focuses on designing systems that automatically learn from data and find hidden characteristics. It can be categorized into supervised, unsupervised, and semi-supervised learning approaches. Supervised learning algorithms, such as Support Vector Machines (SVM) and Random Forest, are used in Network Intrusion Detection Systems (NIDS) research due to their powerful classification power and practicality in computation. Unsupervised learning algorithms, such as feature reduction techniques and clustering techniques, are used to model the fundamental structure or distribution in the data to predict unknown data. Semi-supervised learning, on the other hand, uses unlabeled data for training, suitable for situations where large amounts of labeled data are unavailable.

Deep learning algorithms, which utilize abundant, affordable computation, are applied to various domains, such as visual object recognition, object detection, and network intrusion detection. Convolutional neural networks (CNN) are commonly trained under supervision and are now the benchmark model for computer vision purposes.

Intrusion detection systems (IDS) are crucial in various fields, including the Internet of Things (IoT), web, wireless, and cloud computing. IoT devices require robust IDS solutions to mitigate the diverse array of threats originating from various networks. Numerous algorithms and techniques, including machine and deep learning approaches, play an important role in identifying different types of threats in IoT applications.

In recent years, the importance of intrusion detection systems (IDS) in various domains has increased, including smart homes, cities, industrial settings, buildings, retail environments, and traffic management systems. IDS is crucial for capturing and mitigating emerging threats in these environments. Web applications, such as shopping, banking, and social communication, require security to protect against various types of malicious attacks.

To protect digital data from intrusion, there are three main methods: knowledge about security attack types, security mechanisms, and security services. Security attacks can be classified into active and passive attacks. Security mechanisms include firewalls, which prevent attacks from outside by denying attempts to contact an unauthorized person, and encryption processes that disguise messages.

Intrusion detection systems (IDS) are strong security tools that diagnose and react to attacks, monitoring traffic and malicious attack prevention. Signature-based detection and anomaly-based detection are popular methodologies used for intrusion detection. Signature-based detection involves matching signatures to observable events to identify potential incidents, while anomaly-based detection consists of parameterization, training, and detection. Techniques used for anomaly detection include statistical-based techniques, knowledge-based techniques, and machine learning and deep learning techniques.

Machine learning algorithms have been investigated for intrusion detection systems applications, including support vector machine (SVM), naïve Bayes, genetic algorithm (GA), k-nearest neighbor (K-NN), decision tree (DT), fuzzy logic, and artificial neural networks (ANN). Recently, researchers have used machine learning algorithms for detecting anomalies in traffic datasets with imbalanced class distributions.

Several studies have been conducted in the area of intrusion detection, using various methods and algorithms. Peng et al. (2019) developed a network intrusion detection system using a deep belief neural network (DBN) and a back propagation (BP) neural network classifier for feature extraction. The study found that a Deep Belief Networks (DBN)-based feature learning algorithm was a better solution for feature learning tasks in high-dimensional datasets.

Parampottupadam and Moldovann (2018) proposed a novel approach for real-time intrusion detection systems (IDS) using cloud infrastructure with deep and machine learning methods. They found that using deep learning for binomial and multinomial classification results in the highest accuracy in detecting intrusions while also providing a faster training model for intrusion detection. Wei et al. (2019) examined the utilization of a deep belief network as the foundation for an optimization method, which resulted in an accuracy rate of 83.86%.

Al-Emadi et al. (2020) developed an intelligent system aimed at detecting cyber-attacks using deep learning techniques, notably Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN). The study used deep learning algorithms like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) to develop a resilient intrusion detection system.

Salih et al. (2021) investigated the deployment of deep learning approaches in intrusion detection, finding superior performance and enhanced capability in handling intricate large data sets compared to conventional machine learning techniques.

Moustafa and Slay (2015) introduced the UNSW-NB15 network data collection, which serves as a complete resource for network intrusion detection systems. Alom and Taha (2017) introduced a technique for detecting intrusions in cyber security using unsupervised deep learning algorithms. Shanmugavadivu and Nagarajan (2017) proposed an Intrusion Detection System that utilizes fuzzy logic to efficiently recognize and classify intrusion actions within a network.

Al-Daweri et al. (2020) evaluated the features of the KDD99 and UNSW-NB15 datasets, finding that a classification accuracy of more than 84% could be achieved by utilizing certain characteristics.

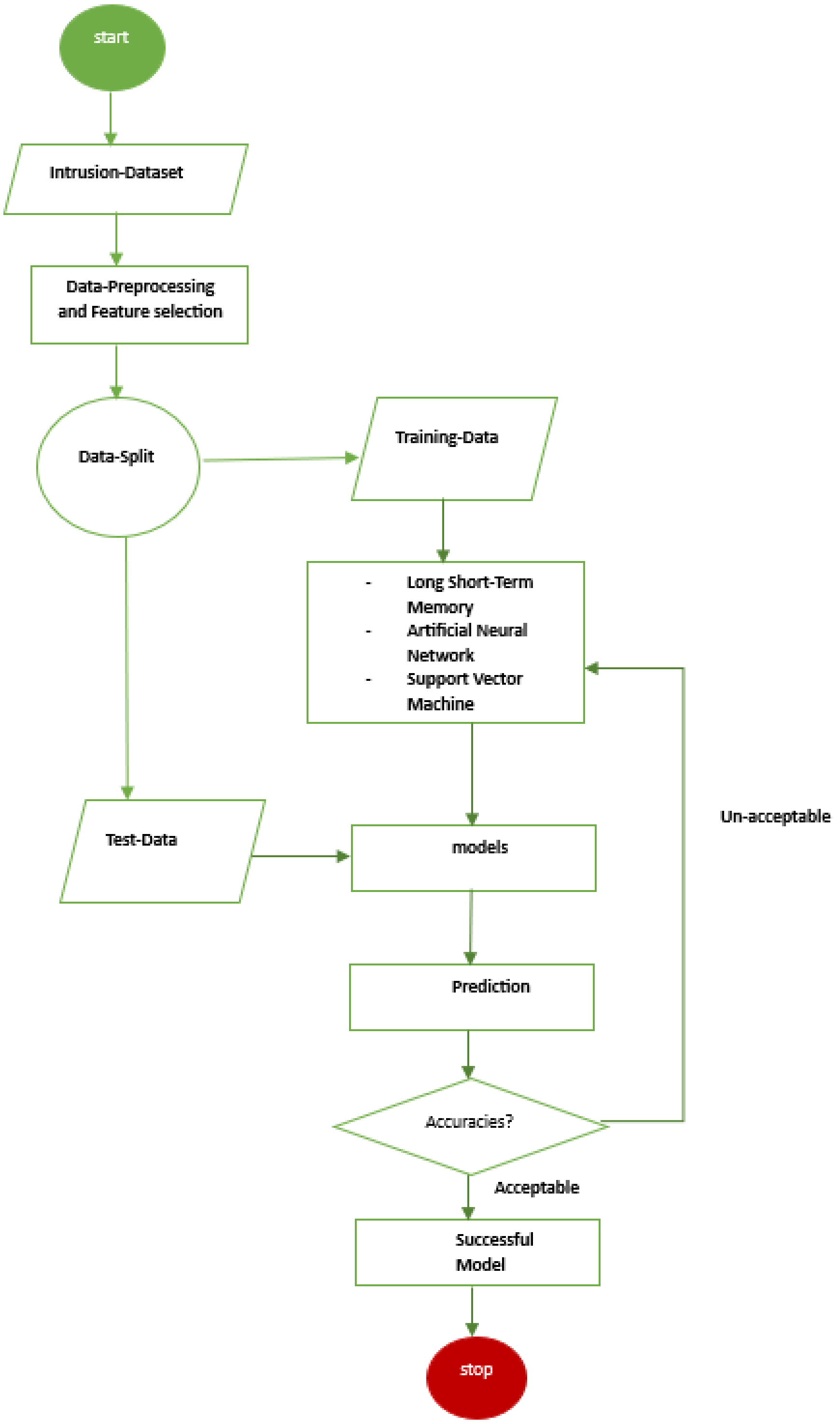
The study by Vinayakumar et al. (2017) and Althubiti et al. (2018) explored the use of convolutional neural networks and Long Short-Term Memory (LSTM) algorithms for intrusion detection. The study used the NSL-KDD dataset and found that CNNs were effective, with a detection rate of 99.79%. Deep Neural Networks (DNN) achieved a 98.90% success rate in accurately detecting the target. The LSTM algorithm was used for anomaly-based intrusion detection, with a satisfactory accuracy of 0.8483, outperforming SVM, MLP, and Naïve Bayes algorithms for multi-classification tasks.

2.1 KNOWLEDGE GAP

The study identifies a knowledge gap in combining deep and machine learning techniques for network intrusion detection. Despite existing studies on CNN, RNN, LSTM, and deep belief networks, there is a lack of discussion on how to integrate multiple tactics to improve performance. Ensemble methods aim to leverage the diversity of models, enhancing dependability and generalization. The study suggests integrating artificial neural networks and support vector machines algorithms and comparing their efficacy to deep learning long short-term memory techniques.

III. Materials And Methodology

This study proposes a three-phase methodology for implementing an intrusion detection system. The first step involves data pre-processing, including missing value handling and noisy data removal. The second step involves feeding the cleansed, scaled, and trained dataset to proposed neural network models, such as artificial neural network, long short-term memory, and support vector machine. The final step involves evaluating the outcomes obtained from the models, with feasible results accepted or repeated until acceptable criteria are met.



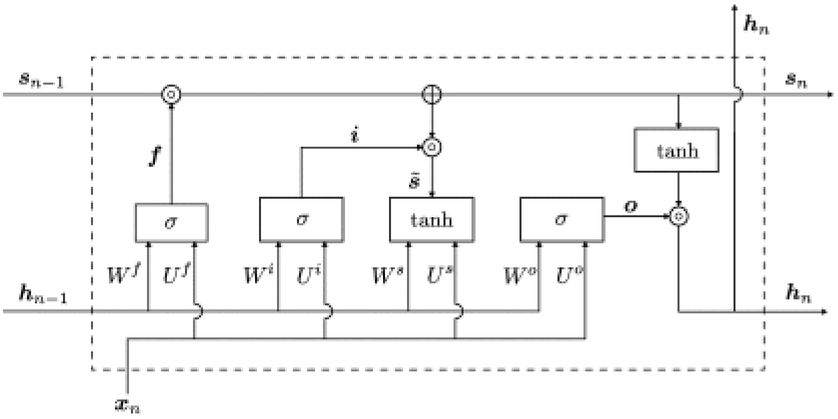
3.1TOOLS AND DESIGN MATERIALS

The proposed network intrusion system will use machine learning models and Python utilities like Python SDK, NumPy, Pandas, Sklearn, matplotlib, and Kera from TensorFlow. The dataset used is the intrusion dataset from the Communications Security Establishment and Canadian Institute for Cyber security, which was abstracted in 2018 to study DDoS data. The dataset's file structure is unbalanced, with eight columns representing a single record in the IDS logging system. Data pre-processing is crucial for improving the accuracy and propensity of the models to predict network intrusions. Pre-processing procedures include label encoding, data scaling, and feature selection to optimize model performance.

3.2 CHARACTERISTICS OF THE ALGORITHMS USED

LSTM

The Long-Short Term Memory (LSTM) is an alternative architectural design for Recurrent Neural Networks that incorporates state management to address the problem of vanishing and exploding gradients. The LSTM networks have an in-built ability to manage data flow through nonlinear elements called 'gates', achieved using the Sigmoid and Rectified Linear Unit Activation Function. The gates are similar to weighting input data, with context influencing the weighting of information. The LSTM is composed of two sets of variables stacked in vectors, s and h. To protect the network's hidden activation layers, three LSTM gates are used: forget, input, and output. The forget gate determines which activations are forgotten and how much, while the input gate affects the proportion of new information added to the protected state.

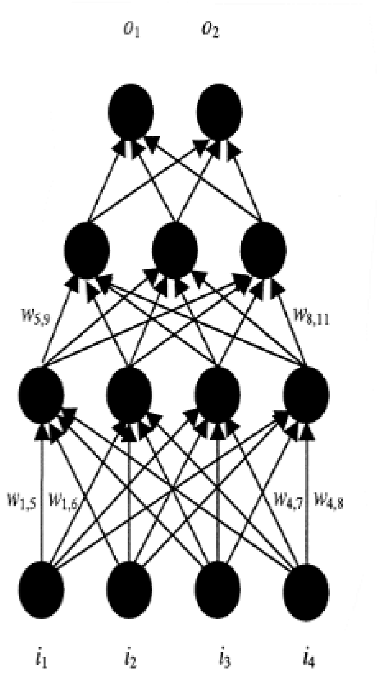


Algorithm 3.1:long short-term memory pseudocode

```
Input: $x = [x_1, x_2, \dots x_n]$
Given parameter: $w_f$, $w_i, w_o, w_c$ $b_f$, $b_i$ $b_0, b_c$
Initialize $b_0, c_0 = \overline{D}$
for $t = 1$ to $n$ do
    calculate $i_t$, (eq 3.1,) $i_t$, (eq 3.2,), $c_t^*$(eq 3.4)
    update cell state $c_t$(eq 3.5)
    calculate $o_t$, (eq 3.3), $h_t$, (eq 3.6)
end for
output $h$= $[h_1, h_2, \dots h_n]$, $h_t \in \mathbb{R}^n$
```

ARTIFICIAL NEURAL NETWORT (ANN)

Artificial neural networks are technologies that mimic the electrical activity of the brain and nervous system by connecting perceptron-like processing units. These networks are organized in layers or vectors, with the output of one layer serving as the input of the next. The weighted data signal through a perceptron mimics the electrical activation of a nerve cell, transferring information within the network or brain. The input values are multiplied by a connection weight, scalar function, and a transfer function is used to produce the output, typically involving the activation function.



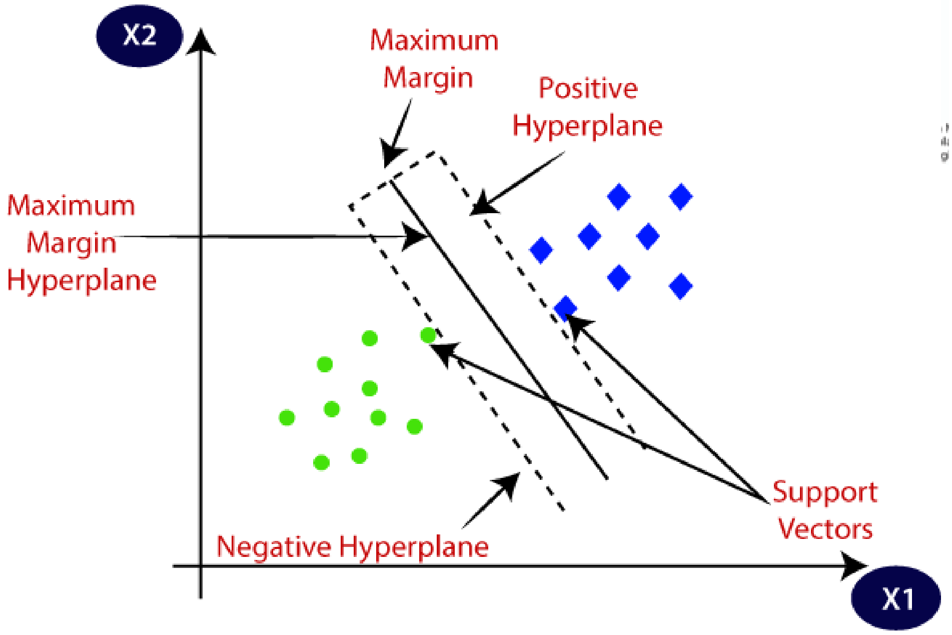
Artificial Neural Network

Algorithm 3.2: ANN Algorithm

```
Step 1: pass the input with some weight to the input layer $(x^1x^2, \dots \dots x^6)$
Step 2: Connect all the inputs to each neuron
Step 3: perform computation at the hidden layers (Eq 3.8)
Step 4: sum all input with their weight
Step 5: Get bias
Step 6: Get the threshold unit
Step 7: Repeat step 3-6 for each of the hidden layers
Step 8: Pass the result to an output layer
```

SUPPORT VECTOR MACHINE (SVM)

The Support Vector Machine (SVM) is a supervised machine learning algorithm used for categorization and regression. It aims to segregate network intrusion into defined classes and find a hyperplane with the largest minimum distance between objects from different signal classes. The proposed SVM uses object signals on the margin edges to separate intrusion classes. The output hyperplane is chosen based on the hyperplane's maximal distance from support vectors. Kernel techniques, including Radial basis function, polynomial kernels, and linear kernels, are used to transform input data into linearly separable data. The algorithm is shown in Algorithm 3.3.



SVM Algorithm

Algorithm 3.3: Support Vector Algorithm

```
Input:
- Training data: a set of pairs $(x_i y_i)$ where $x_i$ is an n-dimensional feature vector and $y_i$ is a binary class label (-1 or +1)
- C: regularization parameter
- Kernel function: a function that computes the dot product of two feature vectors in a high-dimensional space
Output:
- The learned hyperplane that separates the two classes
Step 1:Initialize Lagrange multipliers $alpha_i$ to 0 for all training examples
Step 2:Repeatuntil convergence:
a. For each training $(x_i y_i)$:
i. Compute the margin $M_i = y_i *(w*x_i + b)$, where w is the weight vector and b is the bias term.
    ii. If $alpha_i = 0$ and$y_i * M_i < 1$ or $alpha_i > 0$ and $y_i * M_i! = 1$, then select a second example $(x_j, y_j))$ randomly and perform the following steps:
        1. Compute the kernel function K $(x_i y_i)$
        2. Compute the bounds L and H for the Lagrange multiplier $alpha_j$ using the current value of $alpha_i$and the corresponding labels $y_i$ and y_j
        3. Compute the second derivative of the objective function with respect to $alpha_j$
        4. If the second derivative is negative or zero, set $alpha_j$= H. Otherwise, set $alpha_j$= L if the objective function decreases more with $alpha_j$set to L than with set to H, or set $alpha_j$= H otherwise.
        5. $alpha_j$= changes by more than a small amount, update $alpha_i$= by the same amount to ensure the constraint $sum(alpha_i * y_i)) = 0$ is satisfied.
  b. Update the weight vector w and bias term b using the following formulas:
    w = sum( $sum(alpha_i * (x_i, y_i))$ for all $alpha_i > 0$
    b = $(1/n_s)$ * sum $(y_i$ - sum $(alpha_j * y_j * K(x_j y_j)))$ for all alpha_i > 0, where n_s is the number of support vectors
Step 3:The learned hyperplane is given by (w*x + b) =0
```

3.3 Performance Evaluation Metric Adopted

To test the effectiveness of the adopted models (LSTM, ANN, and SVM). This study proposed some standard evaluation metrics such as accuracy, precision, recall, and F1-score.

i. **Accuracy:**the accuracy evaluation metric determines the proportion of test set inputs that the model properly labels. The accuracy metric can be represented mathematically as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3.12)$$

Where,

The variables TP and TN represent the number of intrusions not correctly predicted as specified in the dataset, the number of intrusion labels that were incorrectly classified as properly allocated from the dataset, and the number of intrusions that were incorrectly classified as allocated but not properly allocated from the dataset where as FP stands for the number of intrusion labels that were incorrectly classed as having been properly allocated from the dataset, and FN stands for the number of intrusion labels that were incorrectly classified as not of intrusion label from the dataset.

ii. **Precision:** describes the proportion of the number of accurately anticipated positive outcomes divided by the total number of expected positive outcomes. Therefore, precision can be represented mathematically as:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3.13)$$

iii. **Recall**: gauges how thorough a classifier is. It is the ratio of the number of positively impacted outcomes from the dataset that were really realized as projected. Recall can be represented mathematically as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3.14)$$

iv. **F-score**: the F-score is a measurement that establishes the harmonic mean of the model's recall and precision and then adds the two values to provide a single score.

$$\text{F}_{\text{Score}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \dots\dots\dots\dots\dots\dots\dots\dots (3.14)$$

IV. RESULTS AND CONCLUSION

This paper presents a practical investigation and implementation of a network intrusion detection model using deep learning algorithms such as LSTM, ANN, SVM, and their hybrid counterparts. The model was developed using the Anaconda programming environment, which is known for its computational robustness and ease of use. Python was chosen as the foundational programming language due to its adaptability, readability, and support for machine learning and data science domains.

Parameter settings for the model include Batch Size, Max Epochs, Optimizers, Loss, Activation Functions, Learning Rate, and Random State. These parameters aim to strike a balance between effective learning, model convergence, and handling the characteristics of the data used in the model.

The Pandas library was used for dataset manipulation, facilitating the ingestion of datasets in various file formats. The dataset used in this study was formatted as a CSV file, and the'read csv' function within the library significantly enhanced the analytical framework of the investigation.

LSTM, ANN, SVM Model Parameter Setting

Parameter	Value
Batch Size	32
Max epochs	10
Optimizer	Rmsprop/Adam
Loss	mean square error
Activation	Relu/Tanh/hard-Sigmoid
Learning Rate	0.005
Random State	42

4.1Models Result

The accuracy report for the network intrusion detection models on the NSL-KDD-99 dataset, as presented in, reveals compelling performance across various algorithms. The Artificial Neural Network (ANN) achieved an outstanding accuracy of 99.9999% after 10 epochs using a batch size of 32 and the Adam optimizer. This result suggests the ANN's proficiency in discerning patterns associated with network intrusions. Support Vector Machine (SVM) model demonstrated high accuracy, reaching 99.93%. SVM, famed for its efficacy in binary classification problems, performed well in detecting network intrusions. The Long Short-Term Memory (LSTM) model, trained for 10 epochs with a batch size of 32 and utilizing the RMSprop optimizer, achieved an accuracy of 99.95%. LSTM, a recurrent neural network variant, proved effective in capturing temporal dependencies within the network traffic data.

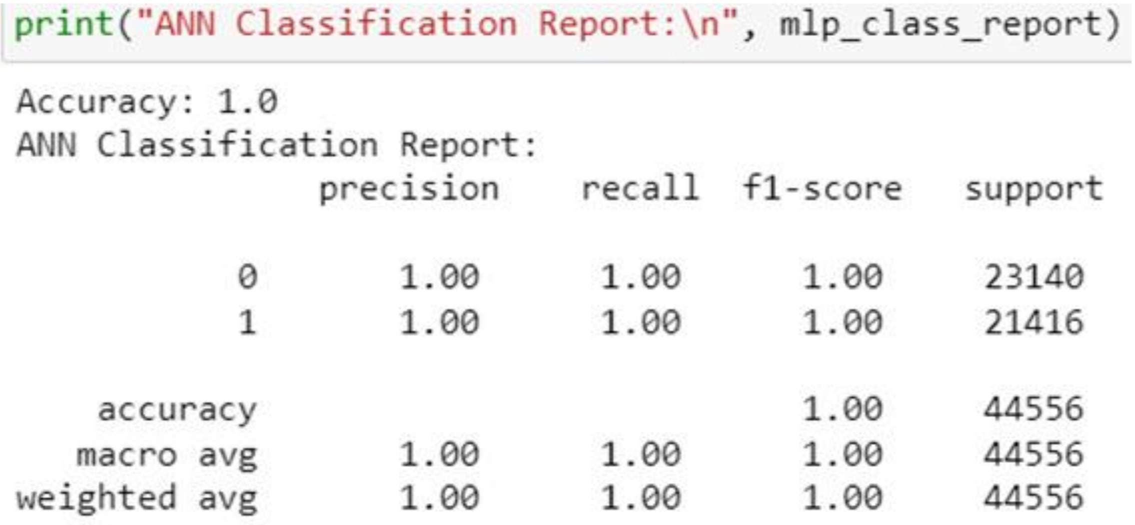
The Hybrid model, integrating ANN, SVM, and LSTM via stacking approach, exhibited exceptional accuracy matching that of the standalone ANN at 99.9999%. This amalgamation of diverse algorithms signifies a synergistic approach, leveraging the strengths of each model to enhance overall intrusion detection accuracy. The consistent high accuracy across all models underscores the strength of the projected methodology in securing network systems against potential threats.

Network Intrusion Models Result

Models	**Epoch**	**Batch**	**Optimizer**	**Accuracy (%)**
ANN	10	32	adam	99.9999
SVM	-	-	-	99.93
LSTM	10	32	rmsprop	99.95
Hybrid	10	32	rmsprop	99.9999

4.2 RESULTS DISCUSSION

The Artificial Neural Network (ANN) model for network intrusion detection is highly effective in classification reports, demonstrating resilience with precision, recall, and f1-score exceeding 1.00 for both normal and invasive classes. The model's high precision and recall values indicate minimal false positives, showcasing its potential for real-world applications.



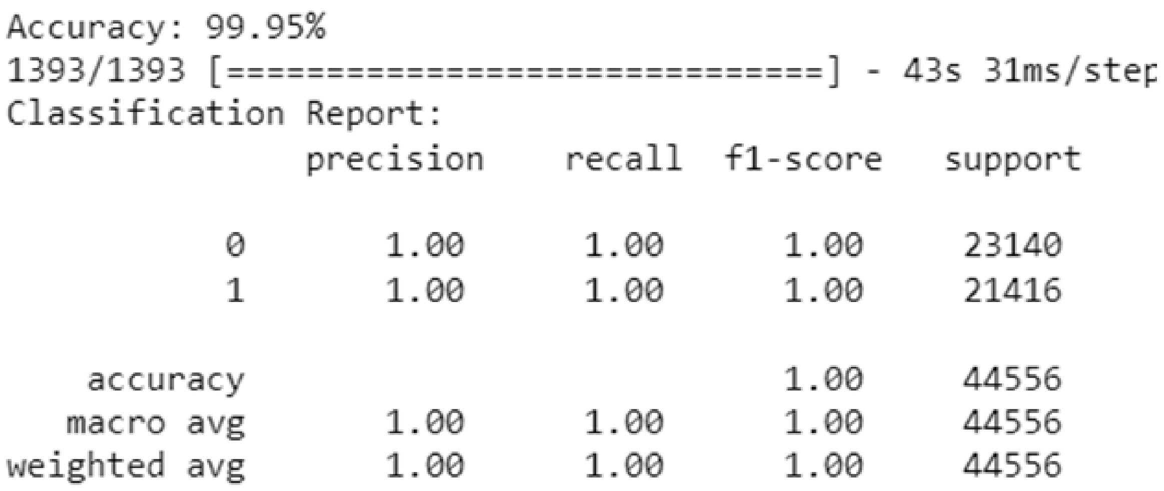
```
print("ANN Classification Report:\n", mlp_class_report)

Accuracy: 1.0
ANN Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00     23140
           1       1.00      1.00      1.00     21416

    accuracy                           1.00     44556
   macro avg       1.00      1.00      1.00     44556
weighted avg       1.00      1.00      1.00     44556
```

The LSTM classification report for network intrusion detection showed exceptional performance in precision, recall, and F1-score metrics, indicating high correctness in identifying normal and intrusive network activities, and demonstrating near-perfect classification accuracy in its application.

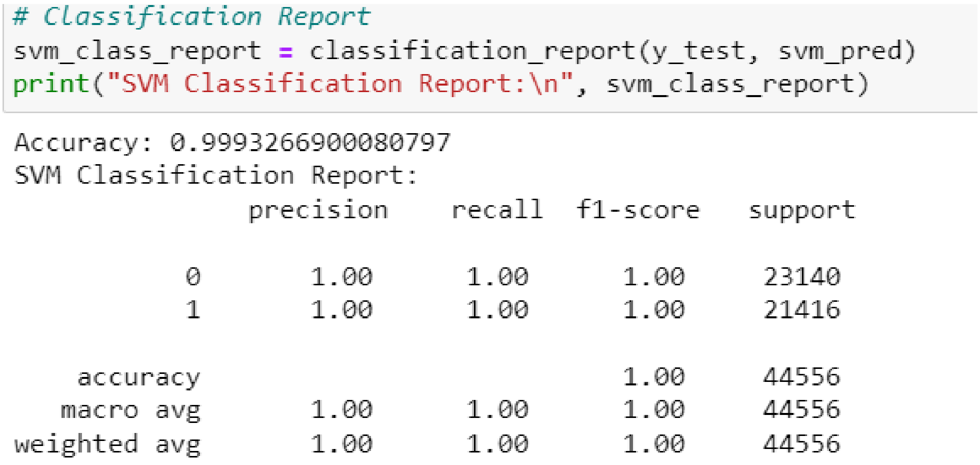


```
Accuracy: 99.95%
1393/1393 [==============================] - 43s 31ms/step
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     23140
           1       1.00      1.00      1.00     21416

    accuracy                           1.00     44556
   macro avg       1.00      1.00      1.00     44556
weighted avg       1.00      1.00      1.00     44556
```

The SVM model for network intrusion detection achieved an impressive 99.93% accuracy, demonstrating high precision in classifying instances as normal or intrusions. Its precision, recall, and f1-score metrics achieved perfect scores of 1.00, ensuring minimal false positives.



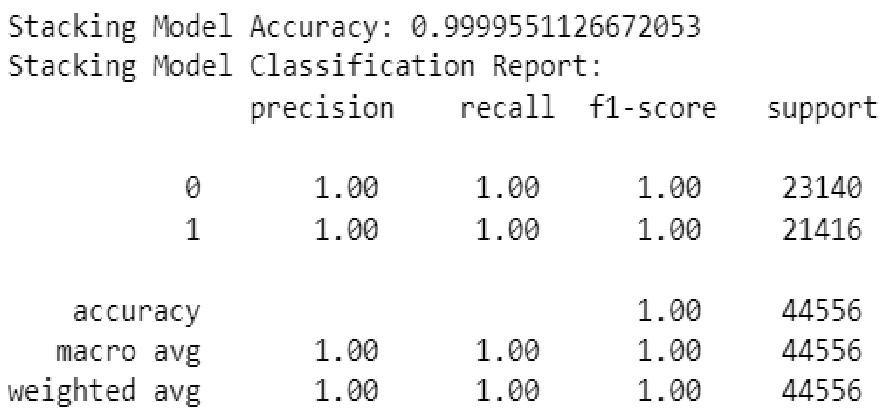
```
# Classification Report
svm_class_report = classification_report(y_test, svm_pred)
print("SVM Classification Report:\n", svm_class_report)
```

```
Accuracy: 0.9993266900080797
SVM Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00     23140
           1       1.00      1.00      1.00     21416

    accuracy                           1.00     44556
   macro avg       1.00      1.00      1.00     44556
weighted avg       1.00      1.00      1.00     44556
```

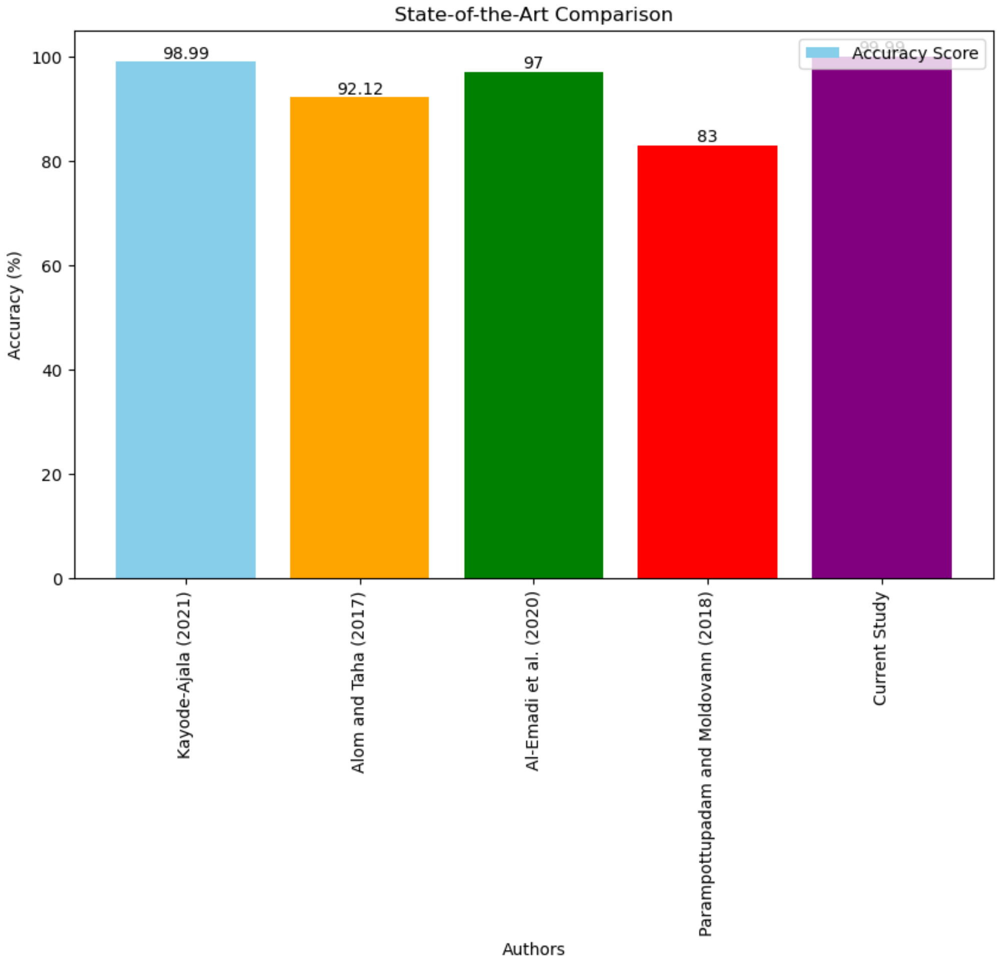
The stacked hybrid model, combining SVM, ANN, and LSTM algorithms, outperforms in network intrusion detection with an accuracy of 99.9955%. The model's precision, recall, and F1-score values are 1.00, demonstrating its robustness in generalizing to unseen data.



```
Stacking Model Accuracy: 0.9999551126672053
Stacking Model Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     23140
           1       1.00      1.00      1.00     21416

    accuracy                           1.00     44556
   macro avg       1.00      1.00      1.00     44556
weighted avg       1.00      1.00      1.00     44556
```



State Of The Art Comparison With Other Works By Authors

V. CONCLUSION

The increasing frequency and sophistication of cyber threats pose a significant challenge to network security. Network Intrusion Detection Models (NIDMs) are crucial for identifying and mitigating potential threats by analyzing network traffic patterns, anomalous behaviors, and known attack signatures. They are essential for safeguarding sensitive information, ensuring data integrity, and maintaining the resilience of interconnected infrastructures. This study developed a systematic three-phase methodology for developing network intrusion detection models, which involved meticulous data preparation, machine learning algorithms, and testing using precision, recall, accuracy, and f1-score metrics. The results showed exceptional performance, with accuracy scores, precision, recall, and f1-score exceeding 99 percent across different metrics. The study suggests future application domains for NIDMs and suggests directions for continuous improvement in the field.

Implementing an intrusion detection system in enterprise networks can improve cybersecurity by proactively detecting and mitigating potential intrusions. It also enhances security in critical infrastructure sectors like energy, transportation, and healthcare. Integrating the system into cloud environments ensures continuous monitoring and protection of cloud-based applications. Future studies should explore ensemble approaches, combining the strengths of different models, and dynamic adaptability, such as reinforcement learning, to improve the system's ability to recognize novel intrusion patterns.

REFERENCES

[1] nce onAburomman, A. A., and Reza M. (2016). Survey of learning methods in intrusion detection systems. International confere advances in electrical, electronic and system Engineering (ICAEES),Putrajaya, pp 362–365. https://doi.org/10.1109/ICAEES.2016. 7888070

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

[3] Ajdani, M., &Ghaffary, H. (2021). Design network intrusion detection system using support vector machine. International Journal of Communication Systems, 34(3), e4689.

[4] Al-Daweri, Muataz& Zainol Ariffin, Khairul Akram & Abdullah, Salwani& Senan, Mohamad. (2020). An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system. Symmetry. 12. 1666. 10.3390/sym12101666.

[5] Aldweesh, A., Derhab, A., and Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowledge-Based Systems, 189, 105124.

[6] Al-Emadi, S., Aisha, A., and Felwa, A., (2020). "Using deep learning techniques for network intrusion detection." In 2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT), pp. 171-176. IEEE,.

[7] Alkhodari, M., andFraiwan, L. (2021). Convolutional and recurrent neural networks for the detection of valvular heart diseases in phonocardiogram recordings. Computer Methods and Programs in Biomedicine, 200, 105940.

[8] Alom, M. Z., and Taha, T. M. (2017, June). Network intrusion detection for cyber security using unsupervised deep learning approaches. In 2017 IEEE national aerospace and electronics conference (NAECON) (pp. 63-69). IEEE.

[9] Althubiti, S. A., Jones, E. M., and Roy, K. (2018, November). LSTM for anomaly-based network intrusion detection. In 2018 28th International telecommunication networks and applications conference (ITNAC) (pp. 1-3). IEEE.

[10] Atkinson R. C., Bellekens, XJ., Hodo, E., Hamilton, A., and Tachtatzis C. (2017). Shallow and deep networks intrusion detection system: a taxonomy and survey. CoRR, arXiv preprint arXiv:1701.02145.2017 Jan 9 Survey of Current Network Intrusion Detection Techniques https://www.cse.wustl.edu/~jain/cse571-07/ftp/ids/. Accessed 26 June 2017

[11] Chen, C., Gong, Y., and Tian, Y. (2008). Semi-supervised learning methods for network intrusion detection. Int Conf Sys, Man Cybern, IEEE.https://doi.org/10.1109/ICSMC. 2008.4811688

[12] Eid, HFA., Darwish, A., Hassanien, A. E., and Abraham A. (2010). Principal components analysis and support vector machine-based intrusion detection system. International conference intelligent systems design and applications (ISDA). 3(1)

[13] Hasan, B. M. S., & Abdulazeez, A. M. (2021). A review of principal component analysis algorithm for dimensionality reduction. Journal of Soft Computing and Data Mining, 2(1), 20-30.

[14] Haweliya J, and Nigam B. (2014). Network intrusion detection using semi supervised support vector machine. Int J Comput Appl. 85, 9

[15] Jian, P., Upadhyaya, S.J., Farooq, F., Govindaraju, V. (2004). "Data mining for intrusion detection: techniques, applications and systems ", in Proceedings of the 20th International Conference on Data Engineering, pp: 877 – 87.

[16] Kayode-Ajala, O.,(2021).Anomaly Detection inNetworkIntrusionDetection Systems usingandMachineLearningand DimensionalityReductionSSRAML.SageScience,4(1), 12–26.

[17] Kizza, J. M. (2024). System intrusion detection and prevention. In Guide to computer network security (pp. 295-323). Cham: Springer international publishing.

[18] Li, X., Peng, L., Yao, X., Cui, S., Hu, Y., You, C., & Chi, T. (2017). Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. Environmental pollution, 231, 997-100

[19] Moustafa, N., and Slay, J. (2015, November). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS) (pp. 1-6). IEEE.

[20] Niyaz, Q., Sun, W., Javaid, AY., and Alam, M. (2016). A deep learning approach for network intrusion detection system. International conference wireless networks and mobile communications (WINCOM).

[21] Nour, B., Pourzandi, M., &Debbabi, M. (2023). A survey on threat hunting in enterprise networks. IEEE Communications Surveys & Tutorials.

[22] Peng, W., Kong, X., Peng, G., Li, X., and Wang, Z. (2019, July). Network intrusion detection based on deep learning. In 2019 International Conference on Communications, Information System and Computer Engineering (CISCE) (pp. 431-435). IEEE.

[23] Pfleeger, C. F., and Pfleeger, S. L. (2003). Security in computing (3rd ed.). Upper Saddle River, NJ: Pearson Education.

[24] Salih, A., Ameen, S. Y., Zeebaree, S. R., Sadeeq, M. A., Kak, S. F., Omar, N., and Ageed, Z. S. (2021). Deep learning approaches for intrusion detection. Asian Journal of Research in Computer Science, 9(4), 50-64.

[25] Shanmugavadivu R. and Nagarajan N. (2017). Network Intrusion Detection System Using Fuzzy Logic. Indian Journal of Computer Science and Engineering (IJCSE) Vol 2(1). p101 – 111. ISSN : 0976-5166

[26] Shun, J., and Malki, H. A. (2008, October). Network intrusion detection system using neural networks. In 2008 fourth international conference on natural computation (Vol. 5, pp. 242-246). IEEE.

[27] Sultana, N., Chilamkurti, N., Peng, W., and Alhadad, R., (2019). Survey on SDN based network intrusion detection system using machine learning approaches. Peer-to-Peer Networking and Applications, 12, 493-501.

[28] Susan, M., Bridges and Rayford B.Vaughn. (2000). "Fuzzy Data Mining and Genetic Algorithms. Applied To Intrusion Detection", In Proceedings of the National Information Systems Security Conference (NISSC), Baltimore, MD, pp.16- 19, October 2000

[29] Tari, Z., Sohrabi, N., Samadi, Y., &Suaboot, J. (2023). Data Exfiltration Threats and Prevention Techniques: Machine Learning and Memory-based Data Security. John Wiley & Sons.

[30] Thakkar, A., &Lohiya, R. (2022). A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. Artificial Intelligence Review, 55(1), 453-563.

[31] Thaseen, S., Kumar Ch. (2013). An analysis of supervised tree-based classifiers for intrusion detection system. In: Proceedings of the international conference on pattern recognition, informatics and mobile engineering (P RIME). Pp. 21–22

[32] University of New Brunswick (2018). IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018). https://www.unb.ca/cic/datasets/ids-2018.html

[33] Vinayakumar., R., Soman, K. P., and Poornachandran, P. (2017, September). Applying convolutional neural network for network intrusion detection. In 2017 International