International Journal of Business Quantitative Economics and Applied Management Research

Volume-7, Issue-2, 2022 ISSN No: 2349-5677

ENHANCING CLOUD DATABASE CYBERSECURITY WITH MACHINE LEARNING-ENABLED INTRUSION DETECTION AND PREVENTION SYSTEMS

Akshay Rajshekar Shiraguppi
Irvine, California
akshayrs1993@gmail.com

Abstract

In this research focused on intrusion detection and prevention gains urgency because of the rising dependency on cloud databases as it strengthens cybersecurity concerns. The research introduces an IDPS system that advances cloud database protection through machine learning methods. The methodology uses CICIDS2017 as its evaluation benchmark to analyze network intrusion detection models. A methodological approach contains three sections that begin with data preprocessing followed by feature selection, then end with classification through Recurrent Neural Network (RNN) and AdaBoost and Naïve Bayes (NB) models. Experimental tests show that RNN outperforms other models with 98% accuracy compared to AdaBoost at 81.83% and Naïve Bayes at 79.99%. The RNN model demonstrates strong accuracy by successfully identifying temporal patterns and complicated attack patterns found in network traffic data. Deep learning techniques show their strength in cloud database security improvement because they detect and block cyber threats during real-time operations. The research investigates the relationship between complex models and computational speed in IDPS solutions specifically for cloud computing systems and provides important direction for designing dependable and expandable security solutions.

Keywords: Intrusion detection, Cloud database, Cybersecurity, Prevention, Machine learning, Artificial intelligence.

I. INTRODUCTION

The adoption of hybrid cloud systems has become widespread because cloud technologies and databases have boosted organizations toward integrating public cloud flexibility with private cloud security and scalability. With the expansion of cloud database usage[1][2], System security measures are essential in protecting confidential data and privacy at all times[3]. Different network protection strategies, including firewall protection policies and antivirus software, have become widely deployed to resolve security concerns. These technologies work at two levels to secure both client system infrastructure and sensitive organization data[4].

However, increasingly sophisticated technologies are required to identify and react to such threats in actual time because cyberattacks are becoming more sophisticated [5][6]. One such instrument that helps detect network or system abuses or intrusions, alert administrators, and record instances for further analysis is an intrusion detection system (IDS)[7]. IDS makes it possible to handle suspicious activity during harmful breakouts without interfering with regular operations[8]. These systems need to be flexible and able to react to new attack techniques as cybersecurity threats continue to change.

The security field requires Intrusion Detection and Prevention Systems (IDPS) as primary defense modules [9][10]. An IDPS continuously detects network and system activities to determine if malicious events happen then it initiates immediate responses to reduce potential threats[11]. The escalating value of information drives malicious actors to execute many types of attacks to obtain important data[12]; implementing an effective IDPS has become even more important because of this situation of escalating cyberattacks[12]. Machine learning and artificial intelligence have become effective methods for tackling these issues. One of the most important fields of study within AI[13][14] allows for the automatic detection and response to threats based on patterns found in system activity, user behavior, and network traffic, thereby enhancing cybersecurity[15].

The potential of artificial intelligence (AI) to improve cybersecurity frameworks, such as malware detection, network traffic analysis, intrusion detection, and social engineering threat identification, has drawn a lot of interest [16][17]. Deep learning and reinforcement learning-based AI-powered systems, in particular, offer strong and flexible security features that can keep up with ever-more-advanced threats. In categorization issues like intrusion detection, machine learning approaches are especially useful [19]; users can benefit from these systems through their ability to handle spam detection and malware analysis features while performing effectively as components of an IDPS framework.

Through machine learning integration in cybersecurity solutions, the development of strong and efficient systems occurred which detect security threats both quickly and accurately.

A. Aim and Contribution

The aim of this Research goals focuses on creating an IDPS through machine learning to boost cloud database cybersecurity effectiveness by detecting and blocking security intrusions better. In addition to training and evaluating RNN, Adaboost, NB, ML and DL models, the study uses the CICIDS2017 dataset for preprocessing tasks. The main mission focuses on creating a data-driven cybersecurity system that optimally secures cloud databases from malicious activities. This paper delivers three main contributions:

- To train and test, the CICIDS2017 dataset is used, providing a comprehensive and representative data source for network intrusion detection in cloud environments.
- Data preprocessing includes various steps for dealing with missing values, reducing data, and removing outliers, followed by categorical feature conversion to prepare the data for dependable model training, one-hot encoding and min-max normalization.
- The detection of intrusions in cloud databases gets evaluated using RNN and Adaboost along with NB, among other machine and DL models.
- The models' efficacy in preventing and detecting intrusions is assessed using key metrics, including recall, accuracy, precision, F1-score, and loss.

- The project results in a machine learning-powered data-driven cybersecurity framework that enhances cloud database intrusion prevention and detection, thus supporting cloud database security development.

B. Structure of the Paper

The study is organized as follows: Section II examines applicable research on machine learning and cloud database cybersecurity. The resources and methods used are described in Section III. The experimental findings of the suggested system are shown in Section IV. Section V, which summarizes the study's main conclusions and insights, comes to a close.

II. LITERATURE REVIEW

This section reviews selected articles on Intrusion Detection Systems in cloud database security analysis using machine learning approaches. Table I summarizes the papers, approaches, data, important conclusions, and noted restrictions or areas for further study.

Freitas De Araujo-Filho et al. (2021) The experiment's findings show that, given the types of attacks taken into consideration, their suggested detection approach achieves detection durations less than 80µs, accuracy more than, and F1-scores higher than 97%. Additionally, when used with a cheap Raspberry Pi, it is the only solution that can eliminate attacker frames before harm is done, in contrast to four cutting-edge intrusion detection systems. Given that one of the main concerns of the automobile sector is cost, such a low-cost deployment is especially desired[18].

Yedukondalu et al. (2021) IDS scan the requested data; if it discovers any harmful material, the request is dropped. These algorithms have employed chi-squared and correlation-based feature selection strategies to reduce the dataset by eliminating superfluous information. The preprocessed dataset is used to train and assess the models, producing notable results in terms of predicted accuracy. The NSL KDD dataset has been used for the testing. Finally, the accuracy of the ANN methodology was 97%, whereas the SVM method's accuracy was around 48%. On this dataset, the ANN model is now outperforming the SVM[19].

Krishna et al. (2020) concentrated on putting in place a DL-based intrusion detection and prevention system that can quickly identify and stop DOS, Probe, R2L, and U2R assaults. The kddcup99 dataset was used to train the Multi-Layer Perceptron DL model, which is highly effective at identifying intrusions as they happen. To identify an attack, the relevant network data is collected, saved as DL model, which is set up to predict assaults in real time, is given a CSV file. The intrusion is stopped at the second step by a script that runs in the background[20].

Sharma, Zavarsky and Butakov (2020) The generated traffic transmitted to an e-commerce web application is included in the CSIC 2010 HTTP dataset. Their test findings show that all evaluated machine learning algorithms perform better in detecting and classifying web-based threats When using the recommended refined feature set extraction. The efficacy of the ML system in attack detection was evaluated using the precision, recall, accuracy, and F-measure metrics. Out of the three algorithms that were studied, the J48 decision tree method had the highest True Positive rate, Precision, and Recall[21].

Nayak et al. (2019) The invader is tracked by utilizing the SORT method in real-time. The NVIDIA Jetson TX2 development platform is also used to build and evaluate the created system for live video streaming, with an average frame rate of 30 and 97% accuracy. Using the reference (beginning) frame and the list of object classes that have been taught, the user may choose the zone of interest (the region that should be free from intrusions) of any size and form, as well as prospective invaders like a person, car, etc. The general character of IDS[22].

Srivastava, Agarwal and Kaur (2019) To recognize these novel forms of attack, detection techniques must be improved. In order to identify irregularities in network traffic, researchers have thoroughly studied machine learning methods. The public repositories now have additional datasets accessible. In this work, In order to find odd patterns in the newly released dataset, they used novel feature reduction-based machine learning approaches. An exceptional 86.15 percent accuracy rate has been attained[23].

TABLE I. A SUMMARY OF INTRUSION DETECTION STUDIES IN CLOUD DATABASE SECURITY USING MACHINE LEARNING

Authors	Methods	Dataset	Key Findings	Limitation & Gap
Freitas De Araujo-Filho et al. (2021)	Detection mechanism using a novel framework for attack detection	Kaggle dataset	The system achieved success rates exceeding 97%, together with F1-score performance at 97% and response times under 80 µs. The system proved effective in eliminating attacking frames for undamaged protection of a low-cost Raspberry Pi platform.	Deployment limited to low-cost platforms; does not mention scalability to more complex systems.
Yedukondalu et al. (2021)	IDS employing chi-squared and correlation-based feature selection techniques	NSL KDD dataset	ANN algorithm achieved 97% accuracy, outperforming SVM (48%) on the dataset.	The performance is based only on the NSL KDD dataset; generalization to other datasets is unclear.
Krishna et al. (2020)	Deep Learning using Multi-Layer Perceptron for attack detection and prevention	KDDCUP99	Achieved high accuracy for attack detection (DOS, Probe, R2L, U2R) and prevention using a background script.	Focuses on KDDCUP99 dataset; does not address scalability or robustness in real-world deployments.
Sharma, Zavarsky, and Butakov	Fine-tuned feature set extraction for	CSIC 2010 HTTP dataset	In terms of Real Positive rate, accuracy, and recall for web-based attack	Does not explore other machine learning algorithms or real-time

(2020)	machine learning algorithms		detection, the J48 decision tree method performed better than the others.	performance.
Nayak et al. (2019)	Real-time tracking using the Simple Online and Real-time Tracking (SORT) method	Kaggle dataset	The system tracked intruders in real-time at 30 frames per second with 97% accuracy through implementations on NVIDIA Jetson TX2. The system maintains the ability to track multiple objects consisting of either vehicles or people.	Limited to video stream tracking; may not generalize to non-video-based intrusion detection.
Srivastava, Agarwal, and Kaur (2019)	Machine learning-based anomaly detection with feature reduction algorithms.	Recently provided dataset	Achieved 86.15% accuracy using novel feature reduction techniques for anomaly detection in network traffic.	Performance might not hold across diverse datasets or attack types. May require further optimization.

III. METHODOLOGY

This research aims to improve cloud database cybersecurity by implementing IDPS systems empowered by machine learning algorithms. The research targets the CICIDS2017 dataset as its foundation to build intelligent threat detection systems that protect cloud environments while minimizing their security weaknesses. Data collection starts the process before moving to extensive preprocessing work, which includes managing missing values and reducing data while eliminating outliers and converting categorical features. The data preprocessing goes through two stages, such as one-hot encoding and min-max normalization, after which feature selection techniques allow for improved model performance. The data undergoes a split procedure, with 30% designated for testing and 70% for training. Adaboost, Naïve Bayes, and RNN combine the processed data used to train DL and ML models. Accuracy, precision, and recall are used to gauge how successful intrusion detection is, and model performance is assessed using the F1-score and loss evaluation. The result represented in Figure 1 forms the basis of an optimized cybersecurity framework that uses data to identify and stop intrusions targeting cloud database systems.

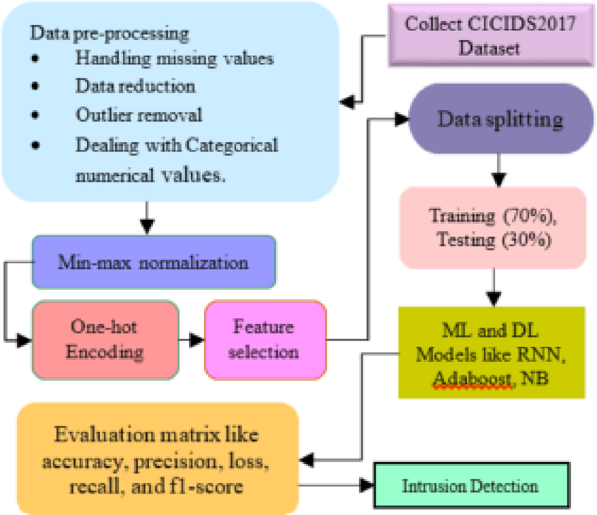
Collect CICIDS2017 Dataset
Data pre-processing
Handling missing values
Data reduction
Outlier removal
Dealing with Categorical numerical values.
Data splitting
Training (70%), Testing (30%)
Min-max normalization
One-hot Encoding
Feature selection
ML and DL Models like RNN, Adaboost, NB
Evaluation matrix like accuracy, precision, loss, recall, and f1-score
Intrusion Detection

Fig. 1. Proposed flowchart for intrusion detection based on cybersecurity.

The proposed methodology of intrusion detection based on cybersecurity systematically outlined, with each step briefly discussed in the section below:

A. Data Collection

The CIC created the CICIDS2017 dataset, which is a standard for IDS. It includes a variety of assaults, such as DDoS, brute force, botnet, web-based attacks, and network penetration, together with innocuous network traffic. The dataset has 80 variables, including flow length, packet size, protocol type, and byte count, is PCAPs and offers labeled network flow data. CICIDS2017, which is intended to simulate actual network traffic, is a thorough tool for assessing Systems that use ML and DL to detect intrusions. The data visualization graphics are provided below:

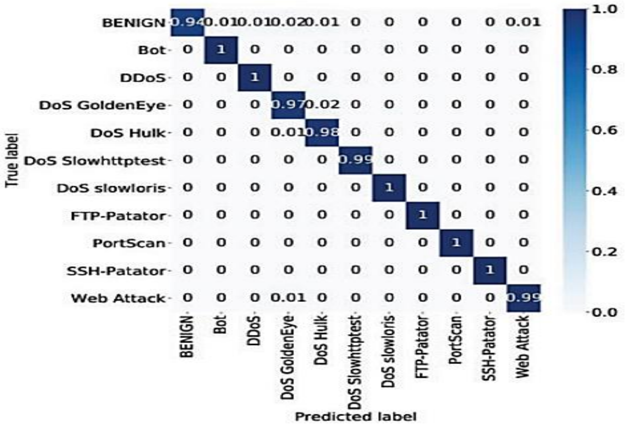

Fig. 2. Heatmap of CICIDS-2017 dataset.

A heatmap of Figure 2 displays the confusion matrix for the CICIDS-2017 dataset's classification performance. High accuracy across the majority of attack types is demonstrated by the diagonal components, which reveal correctly identified cases. Minor misclassifications are observed, particularly in BENIGN. The intensity bar represents classification confidence, ranging from 0 (light) to 1 (dark). Overall, the model efficiently and with few mistakes differentiates between attack categories.

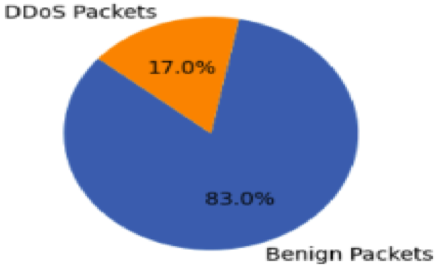

Fig. 3. Pie Chart of Intrusion Detection

Figure 3 presents a pie chart depicting intrusion detection results, categorized into benign and DDoS packets. The chart shows that 83.0% of the data is classified as benign, while 17.0% corresponds to DDoS activity. This visualization highlights the predominance of normal activity alongside a significant presence of malicious, offering insights into the detection system's effectiveness.

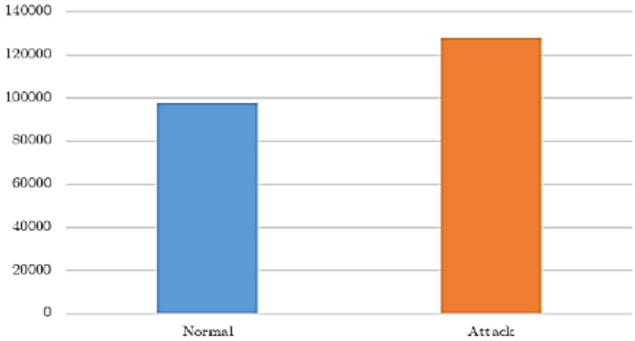

Fig. 4. Bar chart of CICIDS-2017 dataset

In Figure 4 allocation of attack and normal inside the CICIDS-2017 dataset is shown in this bar chart. The X-axis categorizes traffic as either "Normal" or "Attack," while the Y-axis displays the number of occurrences. Attack traffic is more common than regular traffic, as the graphic illustrates, suggesting an unbalanced dataset that is frequently used to assess intrusion detection systems.

B. Data Pre-processing

In order to create ML algorithms and normalize the input for further processing to get the maximum recognition rate, pretreatment is essential. There was several missing, null, and inconsistent values in the datasets, along with duplicate and unnecessary columns that contained no information. Furthermore, the original dataset had three categorical characteristics that were transformed into numerical form using the one-hot encoding technique. To guarantee that the numerical characteristics were converted into a standard format, enhance feature selection, and increase model performance, the dataset is then normalized.

C. Min-Max Normalization

The Min-Max normalization technique is one of numerous methods used in the normalization procedure performed to the numerical characteristics. Improving the system's performance and efficacy requires adjusting all attribute values within a specific range of [0, 1]. Nevertheless, it exhibits unusual affectivity, as in Eq. (1).

$$Z = a - \frac{(\mathrm{xi} - (\min(\mathrm{x}))}{\max(x) - \min(x)} - \min(a) \quad (1)$$

The data element is denoted by xi, the lowest of all data values is min(x), the maximum of all data values is max(x), and Z is a new value. Some variables in the CICIDS 2017 dataset are missing, which leads to errors throughout the normalization procedure. Prior to the normalization procedure, the missing value was handled.

D. One-Hot Encoding

Categorical variables are converted into a numerical format that can comprehend using the appropriate preprocessing techniques, such as one-hot encoding. The approach compares the numerical variable at each level to a fixed beginning point; it is one of the most prevalent.

E. Feature Selection

As part of feature selection, a portion of the initial dataset's relevant characteristics are identified, eliminating irrelevant or redundant attributes to enhance classification performance and reduce memory storage. It mitigates the curse of dimensionality, decreases computational complexity, and improves learning accuracy. Supervised Feature selection techniques fall into three primary categories: wrapper, filter, and embedded models. A widely used filter method is Information Gain, which evaluates attribute significance based on entropy concerning the target class.

F. Data Splitting

The CICIDS2017 study employed a training subset that made up 30% of the entire dataset and a testing selection comprising 70% of the dataset.

G. Classification with Recurrent Neural Network Model

RNNs are so called because they execute the same operation for every element in a sequence, with the result depending on the results of earlier calculations. In RNNs, data can go in either direction. The output of the RNN is used to recycle the input for the following time step. By design, A feedforward neural network consists of an input layer, many hidden layers, and an output layer. An activation function is used to the result after a weight matrix is applied to the inputs of a network node in order to create its output. The network is trained using a backpropagation technique. To get the desired output from the neural network, it is necessary to calculate gradients for each weight and then change each weight individually.

Real-time neural networks (RNNs) have backward connections, wherein the output of one layer gets reinitialized into either that layer or the one before it. For each time step, RNNs use the values computed in the preceding time step to maintain state. A network's equivalent of a short-term memory is this state. Sequential and time series data are commonly represented using regular neural networks (RNNs). To find the RNN's secret states, as Eq. (2):

$$h_t = \sigma(Wx_t + Uh_{t-1} + b_n), fort = T, \dots, 1 \qquad (2)$$

in where σ is a function that detects nonlinearities, xt is a vector representing input at time t, ht is a vector representing current state at time t, W is a matrix representing input to the hidden weight, U is a matrix representing hidden-to-hidden weight, and bh means bias.

H. Evaluation Metrics

F1-score, accuracy, recall, and precision are a few performance metrics. To determine these parameters, use measurements such as TP, FP, TN, and FN. The phrase "FP" describes an intrusion detection system that identifies a "malicious program" as opposed to a "TN" system that detects a "normal program" as such. Similarly, an intrusion detection system detecting a "malicious program" as such is known as a "TP":

The performance measures assessed include the following: Accuracy, Precision, Recall and F1-score and Loss are as follows:

1. Accuracy

The number of accurately recognized instances is determined; it is one of the most significant metrics as Eq. (3):

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+Fp+TN+FN}} \qquad (3)$$

2. Precision

This is known as the "positive predicted value," or the ratio of successfully detected intrusion instances to all projected positive intrusion cases. It is expressed as. It is given by Eq. (4).

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \qquad (4)$$

3. Recall

The "TP" is another name for it, detection rate, or sensitivity," and it is calculated as the percentage of successfully identified intrusion instances relative to the total number of positive intrusion cases. It is given by Eq. (5):

$$Recall = \frac{\text{TP}}{TP+FN} \quad (5)$$

4. F1-score

The F1-score, which is the harmonic mean of accuracy and recall, is a way to calculate it. A more precise count of instances is given by it. that are misclassified from the actual and is formulated as Eq. (6):

$$F1 - Score = \frac{2(Precision * Recall)}{\text{Precision+Recall}} \qquad (6)$$

5. Loss:

The purpose of using an optimizer to modify the weights at each training stage is to lessen the model's loss function during training.

These standards are used to evaluate how well DL and ML models work. F1-Score, accuracy, precision, and recall are used to gauge the models' overall effectiveness.

IV. RESULT ANALYSIS AND DISCUSSION

The results of the experiments conducted on the ML and DL models utilized by cybersecurity threat detection systems are presented in this section. Using a device operating Python 3.3 as a programming language with the Windows 10 operating system, Processor i5 500GB RAM, CPU, and GPU. These metrics are employed to assess performance: Recall, accuracy, precision, and F1-score. Results of the logistic regression model's performance in threat detection are displayed in Table II.

TABLE II. ML AND DL MODELS FOR INTRUSION DETECTION IN CLOUD DATABASE CYBERSECURITY USING THE CICIDS2017 DATASET.

Performance Measures	RNN
Accuracy	98
Precision	98
Recall	98
F1-score	98

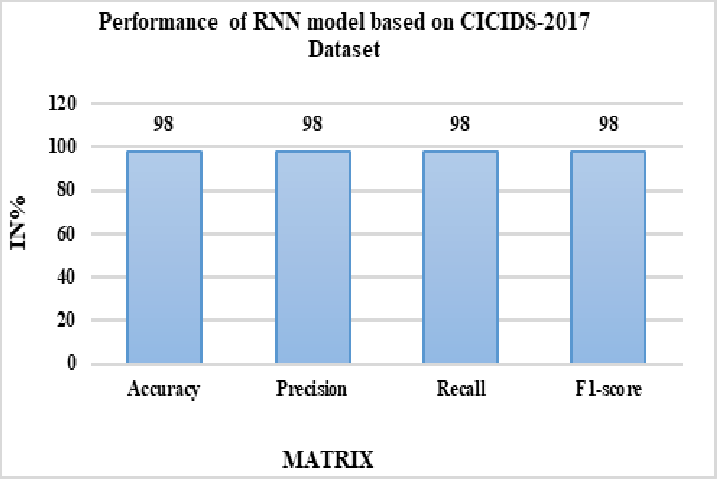

Fig. 5. Bar Graph for RNN Model Performance

Metrics for an RNN model's performance on the CICIDS-2017 dataset are presented in Figure 5 and Table II. Four important performance metrics are graphically shown by the bar chart: 98% was attained in accuracy, F1-score, precision, and recall. The table reinforces these results by presenting the exact numerical values for each metric. The continuously high results on all performance metrics show that the RNN model is very good at finding intrusions in the dataset. Its balanced accuracy and recall result in an ideal F1 score.

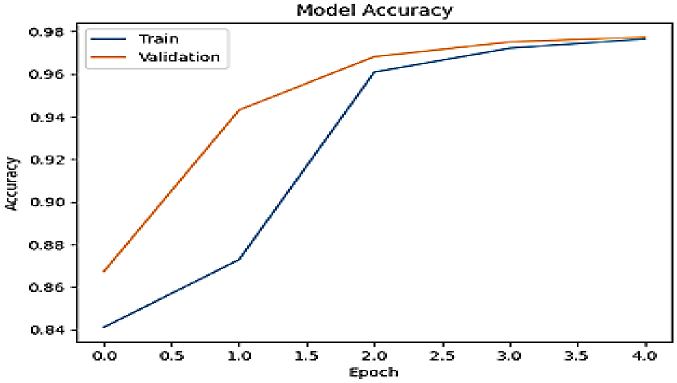

Fig. 6. Accuracy graph for RNN model

In Figure 6, the accuracy graph of the RNN model displays the accuracy of training and validation versus the number of epochs. The graph shows a steady rise in training accuracy from around 0.84 to 0.98, while the validation accuracy also rises, plateauing at approximately 0.98 after an initial steeper climb, suggesting good model fit and generalization with minimal overfitting.

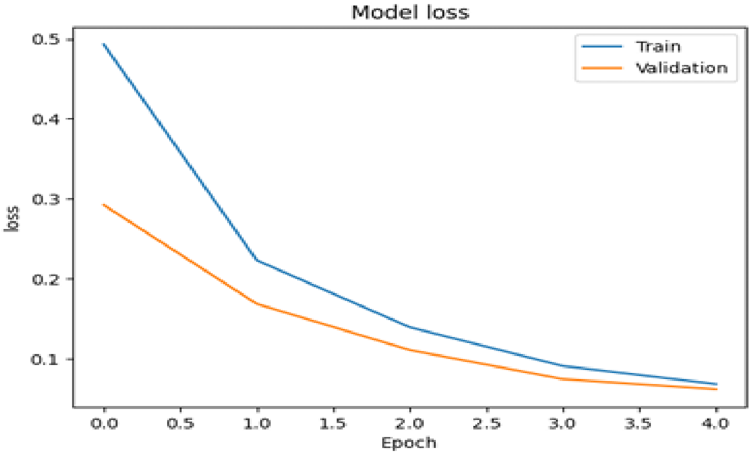

Fig. 7. Loss and validation graph for RNN model

Figure 7, the validation and loss graph of the RNN model, displays decreasing loss for each of the four epochs in the training and validation sets. The blue line represents the training lost, exhibiting a steeper decrease, dropping from approximately 0.5 to below 0.1. The validation loss, shown in orange, also decreases, though at a slower rate, leveling off at around 0.07 after the initial decline, suggesting that learning is taking place in the model effectively and generalizing well to unseen data without significant overfitting.

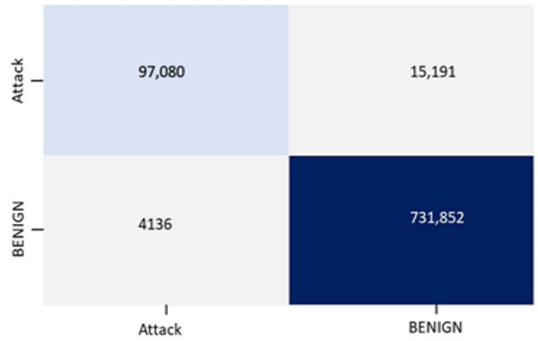

Fig. 8. Confusion Matrix of RNN model

Figure 8 represents the confusion matrix of the RNN model and shows that it correctly classified 97,080 "Attack" instances TP and 731,852 "Benign" instances TN while misclassifying 15,191 benign cases as attacks FP and failing to detect 4,136 actual attacks FN. This indicates strong overall accuracy, with high precision for attack detection and a relatively low false negative rate, suggesting the model effectively differentiates between the two classes. However, the false positives could be a concern, potentially leading to unnecessary alerts.

A. Comparative Analysis and Discussion

A comparative analysis of cloud database cybersecurity using the CICIDS2017 dataset is presented in this section. Comparing ML and DL models like RNN Adaboost [24] and NB[25], as displayed in Table III and according to the f1-score, accuracy, precision, and recall are performance markers.

TABLE III. COMPARISON OF ML AND DL MODELS USING CICIDS2017 DATASET FOR CLOUD DATABASE CYBERSECURITY

Performance Measures	**RNN**	**Adaboost**[24]	**NB**[25]
Accuracy	98	81.83	79.99
Precision	98	81.83	86.03
Recall	98	99	90.06
F1-score	98	90.01	88.06

The quantitative assessment of the CICIDS2017 dataset for cloud database cybersecurity stands in Table III through an analysis comparing ML and DL models. The best threat detection performance is provided by the RNN, which achieves 98% accuracy across all metrics such as F1-score assessments, memory, and accuracy. The detection capabilities of AdaBoost are strong according to its 90.01% F1-Score, but potential false positive outcomes lead to 81.83% accuracy and precision coupled with 99% recall. Naive Bayes produces the lowest result at 79.99% accuracy while it achieves precisions at 86.03% and recalls at 90.06%, and the F1-score reaches 88.06%, demonstrating general classification deficiencies. Cloud database cybersecurity effectiveness reaches its peak with DL-based models, especially RNN because they surpass traditional ML techniques in detection performance.

V. CONCLUSION AND FUTURE WORK

The solution of intrusion detection remains a complex issue which strongly affects service quality together with reliability. Security and reliability, together with performance, experience major negative effects because of the significant difficulty in detecting intrusions in cloud databases. The RNN model is shown to be the best at detecting cybersecurity vulnerabilities in cloud databases through analysis of data from the CICIDS2017 dataset. This model outperformed traditional ML models like AdaBoost (81.83% accuracy) and Naïve Bayes (79.99% accuracy) with 98% accuracy and F1-score, recall, and precision. The strong classification ability of the system is reflected in the confusion matrix that correctly classifies most attack instances and benign cases together with some remaining false positive results. These results highlight that DL, particularly RNNs, is the most effective approach for cybersecurity threat detection.

Future objectives aim to improve intrusion detection by incorporating additional DL architectures, such as LSTM and Transformer models, to enhance sequential learning capabilities. Furthermore, the implementation of hybrid models integrating multiple ML and DL techniques could further optimize detection performance. Adopting advanced feature selection techniques and real-time processing methods will also enhance model efficiency and reduce computational overhead.

REFERENCES

1. A. and P. Khare, "Cloud Security Challenges : Implementing Best Practices for Secure SaaS Application Development," Int. J. Curr. Eng. Technol., vol. 11, no. 6, pp. 669–676, 2021, doi: https://doi.org/10.14741/ijcet/v.11.6.11.
2. O. Zasuhina and K. Saharovskaya, "Databases in Cloud Technologies," Mod. Technol. Sci. Technol. Prog., pp. 115–116, Jun. 2020, doi: 10.36629/2686-9896-2020-1-115-116.
3. P. Verma et al., "A Novel Intrusion Detection Approach Using Machine Learning Ensemble for IoT Environments," Appl. Sci., vol. 11, no. 21, p. 10268, Nov. 2021, doi: 10.3390/app112110268.
4. S. S. S. Neeli, "Optimizing Data Management and Business Intelligence: Integrating Database Engineering with AI-driven Decision Making," Int. J. Commun. Networks Inf. Secur., vol. 17, no. 01, p. 24, 2025.
5. S. S. S. Neeli, "Optimizing Database Management with DevOps: Strategies and Real-World Examples," J. Adv. Dev. Res., vol. 11, no. 1, p. 8, 2020.
6. A. Borkar, A. Donode, and A. Kumari, "A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS)," in Proceedings of the International Conference on Inventive Computing and Informatics, ICICI 2017, 2018. doi: 10.1109/ICICI.2017.8365277.
7. M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices," IEEE Internet Things J., 2020, doi: 10.1109/JIOT.2020.2970501.
8. S. Laqtib, K. El Yassini, and M. L. Hasnaoui, "A technical review and comparative analysis of machine learning techniques for intrusion detection systems in MANET," International

Journal of Electrical and Computer Engineering. 2020. doi: 10.11591/ijece.v10i3.pp2701-2709.

9. S. Chatterjee, “Risk Management in Advanced Persistent Threats (APTs) for Critical Infrastructure in the Utility Industry,” Int. J. Multidiscip. Res., vol. 3, no. 4, pp. 1–10, 2021.
10. N. Mazhar, R. Salleh, M. A. Hossain, and M. Zeeshan, “SDN based intrusion detection and prevention systems using manufacturer usage description: a survey,” Int. J. Adv. Comput. Sci. Appl., 2020, doi: 10.14569/IJACSA.2020.0111283.
11. M. Baykara and R. Das, “A novel honeypot based security approach for real-time intrusion detection and prevention systems,” J. Inf. Secur. Appl., 2018, doi: 10.1016/j.jisa.2018.06.004.
12. K. Alsubhi and H. M. AlJahdali, “Intrusion detection and prevention systems as a service in could-based environment,” Int. J. Adv. Comput. Sci. Appl., 2018, doi: 10.14569/IJACSA.2018.090738.
13. Godavari Modalavalasa, “The Role of DevOps in Streamlining Software Delivery: Key Practices for Seamless CI/CD,” Int. J. Adv. Res. Sci. Commun. Technol., vol. 1, no. 12, pp. 258–267, Jan. 2021, doi: 10.48175/IJARSCT-8978C.
14. S. T. Bakhsh, S. Alghamdi, R. A. Alsemmeari, and S. R. Hassan, “An adaptive intrusion detection and prevention system for Internet of Things,” Int. J. Distrib. Sens. Networks, 2019, doi: 10.1177/1550147719888109.
15. C. Birkinshaw, E. Rouka, and V. G. Vassilakis, “Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks,” J. Netw. Comput. Appl., 2019, doi: 10.1016/j.jnca.2019.03.005.
16. S. S. S. Neeli, “Key Challenges and Strategies in Managing Databases for Data Science and Machine Learning,” Int. J. Lead. Res. Publ., vol. 2, no. 3, p. 9, 2021.
17. I. H. Sarker, M. H. Furhad, and R. Nowrozy, “AI-Driven Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions,” SN Computer Science. 2021. doi: 10.1007/s42979-021-00557-0.
18. P. Freitas De Araujo-Filho, A. J. Pinheiro, G. Kaddoum, D. R. Campelo, and F. L. Soares, “An Efficient Intrusion Prevention System for CAN: Hindering Cyber-Attacks with a Low-Cost Platform,” IEEE Access, 2021, doi: 10.1109/ACCESS.2021.3136147.
19. G. Yedukondalu, G. H. Bindu, J. Pavan, G. Venkatesh, and A. Saiteja, “Intrusion Detection System Framework Using Machine Learning,” in Proceedings of the 3rd International Conference on Inventive Research in Computing Applications, ICIRCA 2021, 2021. doi: 10.1109/ICIRCA51532.2021.9544717.
20. A. Krishna, M. A. Ashik Lal, A. J. Mathewkutty, D. S. Jacob, and M. Hari, “Intrusion Detection and Prevention System Using Deep Learning,” in Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020, 2020. doi: 10.1109/ICESC48915.2020.9155711.
21. S. Sharma, P. Zavarsky, and S. Butakov, “Machine Learning based Intrusion Detection System for Web-Based Attacks,” in 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), IEEE, May 2020, pp. 227–230. doi: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00048.
22. R. Nayak, M. M. Behera, U. C. Pati, and S. K. Das, “Video-based Real-time Intrusion

Detection System using Deep-Learning for Smart City Applications," in International Symposium on Advanced Networks and Telecommunication Systems, ANTS, 2019. doi: 10.1109/ANTS47819.2019.9117960.

23. A. Srivastava, A. Agarwal, and G. Kaur, "Novel Machine Learning Technique for Intrusion Detection in Recent Network-based Attacks," in 2019 4th International Conference on Information Systems and Computer Networks, ISCON 2019, 2019. doi: 10.1109/ISCON47742.2019.9036172.
24. A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset," in Journal of Physics: Conference Series, 2019. doi: 10.1088/1742-6596/1192/1/012018.
25. A. A. Abdulrahman and M. K. Ibrahem, "Evaluation of DDoS attacks Detection in a New Intrusion Dataset Based on Classification Algorithms," Iraqi J. Inf. Commun. Technol., vol. 1, no. 3, pp. 49–55, 2019, doi: 10.31987/ijict.1.3.40.