International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group | Volume 7 Issue 2 PP 1-9, 2026
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V7I2P101

Original Article

Data Analytics Approaches for Effective Threat Identification in Cloud Databases

Mr. Mohit Sahu
Department of Computer Sciences and Applications, Assistant Professor, Mandsaur University, Mandsaur.

Received On: 15/02/2026 ***Revised On: 18/03/2026*** ***Accepted On: 24/03/2026*** ***Published On: 01/04/2026***

Abstract *- The threats are also becoming more sophisticated and multifaceted in the world as the technology advances, including the data breaching, insecure interfaces, shared technology vulnerability, and distributed attacks. One of the most pressing problems in today's IT infrastructure is ensuring the safety and dependability of cloud computing by detecting threats in server databases. This study constructs and evaluates intrusion detection models using the CICIDS2017 dataset, which contains both benign and harmful traffic. Extensive data preprocessing steps were used to improve the dataset's quality and address the problem of class imbalance. Data purification, normalisation, feature selection, and SMOTE data balancing were all part of these processes. Some common measures used for training and testing LSTM models include accuracy (ACC), precision (PRE), recall (REC), and F1-score (F1). In the best trial result, the model achieved an ACC of 98.51%, PRE of 99.0%, recall of 98.0%, and F1 of 99.0%, proving that it is trustworthy in differentiating between benign and malicious traffic. Training and validation curves demonstrate successful learning and high generalization, and there is not a great deal of overfitting. The analysis of the confusion matrices also confirms the high rates of detection of all types of attacks where most of them were over 95 percent correct. The LSTM has an obvious advantage when compared to traditional models (MLP, SVM, XGBoost, DeepGFL): it is more effective, as it can identify intricate temporal patterns in network traffic that cannot be identified by other models. Finally, to improve the efficacy of intrusion detection systems in the modern cloud-based environment, the proposed LSTM model provides a strong and efficient method for detecting threats in cloud databases.*

Keywords *- Cybersecurity, Cloud Computing, Internet of Things (IoT), Threat Detection, Distributed Security, Machine Learning.*

1. Introduction

The phenomenal rise of the internet and other forms of information technology in recent years has altered people's daily lives and places of employment. Digital technology have grown integral to many facets of modern life, including online shopping, social media, telemedicine, and telecommuting. Computer networks are more vulnerable to cyber risks including data breaches, unauthorised access, and other harmful activities as a result of the digital revolution [1]. Governments, businesses, and individuals all over the world are facing severe threats by cybercriminals advancing new methods and systems to take advantage of network vulnerabilities[2][3][4]. The importance of effective intrusion detection systems (IDS) that can identify and counteract intrusions in real time has so grown [5][4]. To detect known threats in a cloud context, traditional intrusion detection systems rely on rule-based or signature-based methods [6][7][8]. Network traffic patterns are compared with a database of predetermined attack signatures or rules of harmful activity in these approaches. Although signature-based IDS have been successful in responding to threats that had already been recognized, detection of new or advanced attacks that are not either in a pattern is not usually successful[9][10]. Further, the rule-based approaches often produce large volumes of false positive results, which raise the burden of operations and efficiency in network security management[11][12].

In recent times ML methods have been highly used to make IDS more accurate, efficient and more adaptable. Machine learning based IDS is dynamically analyzed compared to traditional methods that use rule-based methods, which is dependent on more rigid signatures and fixed sets of rules[13][14]. With these models, it is possible to learn and differentiate the normal and abnormal patterns of behavior on its own and identify the previously unfamiliar threats and adapt to the changing methods of cyberattacks[15][16]. However, because cyber threats [13] are always evolving and network traffic data is complicated, machine learning models aren't always effective [17][18]. Models for intrusion detection supported by deep learning The process of learning representations from raw traffic data might be made simpler. Consequently, DL has immense promise for the advancement of IDS and the reinforcement of cybersecurity measures in modern network architecture.

1.1. Motivation and Contribution

Conventional security systems fail to identify advanced and emerging threats, which may result in the possible breach of data, disruption of services, and loss of finances. This requires creation of smart data-driven IDS that are able to detect known attacks and unknown attacks with high PRE. With reference to datasets like CICIDS2017, which models the real-world attacker cases, and to the high-level methods of increasing the PRE of detection and decreasing the cost of

computing them, etc. The rationale of this work is to create a powerful threat detection framework that can overcome class imbalance, use pertinent features, and powerful models to empower the security of cloud databases [19] and guarantee the dependability of digital infrastructures. The following are the major contributions of this research:

- Implemented systematic cleaning, normalization, and transformation of the CICIDS2017 dataset to ensure data quality and reliability.
- Improved model performance and reduced complexity by applying feature selection approaches to eliminate redundant and irrelevant attributes.
- Used SMOTE to balance the dataset, and the minority attack class is fairly represented, and the model is more likely to generalize.
- Developed and tested an LSTM model that is able to address sequential and temporal dependencies in network traffic to detect threats better.
- Evaluated model behavior holistically based on a variety of assessment measures, such as ACC, PRE, REC, and F1.

1.2. Justification and Novelty

The rationale behind this research is the growing reliance on cloud databases that are extremely susceptible to numerous emerging cyber threats. The use of conventional intrusion detection systems is usually limited by issues like: class imbalance, working with high dimensional data and inability to efficiently extract sequential patterns in network traffic. To address these limitations, this paper makes use of methodical feature selection, data balancing algorithms, and deep learning high-end models to improve the detection, efficiency, and PRE. The originality of the research is the combination of feature optimization with a DL-based method, in particular, the application of LSTM, which can learn temporal correlations and multifaceted relationships between network traffic. In contrast to traditional approaches, the suggested framework does not only enhance performance in detection, it also provides scalability and robustness, which is why its a better solution to protect cloud databases against contemporary security threats.

2. Literature Review

The purpose of this literature review is to summarise and analyse recent research on intrusion detection systems, focussing on how these systems make use of deep learning and machine learning models.

Dhinakaran *et al.* (2024) investigates how powerful machine learning is at protecting data stored in the cloud. Three trials were utilised to assess different machine learning models in this scenario. Using a RF model, the experiment established that it was 95% accurate, with a REC of 0.96, a PRE of 0.92, and an F1 Score of 0.94. That the model achieves a reasonable ratio of accurate to false positives for classifying security threats is demonstrated. Using a DNN improved ACC to 97% in the second experiment. F1, PRE, and REC scores of 0.96, 0.98, and 0.94, respectively, show that the DNN can distinguish between threats and regular behaviour. Due to its excellent pattern recognition capabilities, the model is a useful instrument for cloud security. Experiment 3 introduced Q-learning, a reinforcement learning technique, to the field of security analysis. Given that the model was able to detect threats with an 88% success rate and a false positive rate of only 0.05, a compromise between the two was achieved. A lower false negative rate of 0.12 suggests that threat detection is becoming more accurate [20].

Tiwari and Jain (2022) provide a new firewall method for secure cloud computing settings that makes use of DL and ML. A novel combination methodology known as most frequent decision is employed by the suggested methods to detect and classify incoming packets of traffic. This methodology takes into account both the nodes' past decisions and the present decision of the machine learning algorithm to predict the final strike category. There is an improvement in both learning performance and system correctness using this strategy. An open-source dataset called UNSW-NB-15 is used to get the results. Statistics show that it improves anomaly detection by 97.68% [21].

Garg *et al.* (2022) Look into any inconsistencies discovered with security standards that were once commonly used, such as those for wired or wireless networks. Aside from the connection and traffic patterns at CoT, this data is generated by a multiplicity of sources. This research utilises three key ML algorithms: KNN, CNN, and NB, with the main focus being anomaly identification in the CoT. An examination of anomalies within the BotIoT dataset, which is produced by simulating switches and devices connected to the IoTs. With an F1 of 99.5%, a CNN can reach an ACC of 99.94%. Positive results regarding the RoC curve and confusion matrix are demonstrated by the evaluation [22].

Ntambu and Adeshina (2021) offers a paradigm for proactively tracking and identifying unusual consumption of virtual machine resources. The proposed model can identify such an abnormality and pinpoint when it occurred. A range of virtual machine resource metrics were used to train and evaluate the model on a workload trace sampling from virtual machines. For this, turned to Isolation Forest and OCSVM, two popular SVM approaches. In contrast to Isolation Forest's 0.93 and 0.80 average F1 for hourly and daily time series, respectively, OCSVM attained 0.97 and 0.89. Evidently, the model is amenable to both techniques; nonetheless, OCSVM outperformed Isolation Forest in terms of classification ACC [23].

Yasarathna and Munasinghe (2020) The main focus was on analysing data from cloud networks using one-class classification methods, particularly Autoencoder and OCSVM, to detect abnormalities. The use of YAHOO data for anomaly detection in this work is unprecedented, as far as is known. The results demonstrate that Autoencoder achieves an ACC of 96.02% and OCSVM achieves an ACC of 79.55% in detecting outliers. Utilising the UNSW-NB15 benchmarked data set, further investigation into the effectiveness of a one-class classification method was also

carried out. With an ACC of 99.10%, the Autoencoder outperformed the OCSVM, which only managed 60.89%. Neural network-based approaches perform better than kernel-based methods for anomaly identification in cloud network data, as shown in the results [24].

Salman *et al.* (2017) trained learning models to identify and classify various assaults using a widely used publically available dataset. Have really employed two supervised ML methods, namely RF and LR. demonstrate that similarities across assaults can lead to less accurate classification, even with excellent detection processes. With an ACC rating of 93.6% in classification and a detection rate of almost 99%, the data show that some attacks cannot be classified. Then, should make the case that the same ML methods can be used to classify multi-cloud setups [25].

The literature demonstrates robust performance of ML and DL on IDS, but the majority of methods use static datasets, and therefore, have class imbalance issues, and no time series modeling of network traffic. One-class and reinforcement learning have a tradeoff to false positives, generalization, whereas anomaly detection does not generalize in IoT and VM scenarios (Table 1). Therefore, it is evident that there is a need for a robust, sequence-sensitive framework, such as LSTM, capable of tracking dynamic traffic patterns, balancing patterns across various attack types, and providing a stable level of applicability in real-world cloud database settings.

Table 1: Overview of Recent Study for Threat Detection in Cloud Environment Using Machine Learning

Author	Proposed Work	Dataset	Key Findings	Challenges/recommendation
Dhinakaran *et al.* (2024)	Applied ML for cloud data security using RF, DNN, and Q-learning.	Cloud security datasets	RF: 95% accuracy; DNN: 97% accuracy (best performance); Q-learning: 88% detection with trade-off in false positives.	Reinforcement learning requires tuning to reduce false positives/negatives.
Tiwari & Jain (2022)	Proposed ML + DL firewall mechanism using "most frequent decision" method for packet classification.	UNSW-NB15	Improved anomaly detection with 97.68% accuracy.	Method relies on previous node decisions, requiring optimization for real-time scalability.
Garg *et al.* (2022)	Anomaly detection in Cloud of Things (CoT) using KNN, CNN, and Naive Bayes.	Bot-IoT dataset	CNN achieved 99.94% accuracy and 99.5% F1-score, with superior ROC and confusion-matrix performance.	Further testing needed in real-world CoT environments with diverse traffic.
Ntambu & Adeshina (2021)	Proactive VM anomaly monitoring using Isolation Forest & OCSVM.	VM workload trace data	OCSVM outperformed with F1-score: 0.97 (hourly) and 0.89 (daily); Isolation Forest: 0.93 & 0.80.	Isolation Forest less effective; models require fine-tuning for different workloads.
Yasarathna & Munasinghe (2020)	One-class classification (OCSVM & Autoencoder) for anomaly detection in cloud networks.	YAHOO Synthetic & UNSW-NB15	Autoencoder achieved 96.02% (YAHOO) and 99.10% (UNSW-NB15); OCSVM lower at 79.05% and 60.89%.	Neural networks outperform kernel-based methods; class imbalance still a challenge.
Salman *et al.* (2017)	Attack detection and categorization using Linear Regression (LR) and Random Forest (RF).	Public dataset (not specified, likely KDD or similar)	Detection accuracy >99%, categorization accuracy 93.6%; difficulty in categorizing similar attacks.	Recommends extending classification to multi-cloud environments.

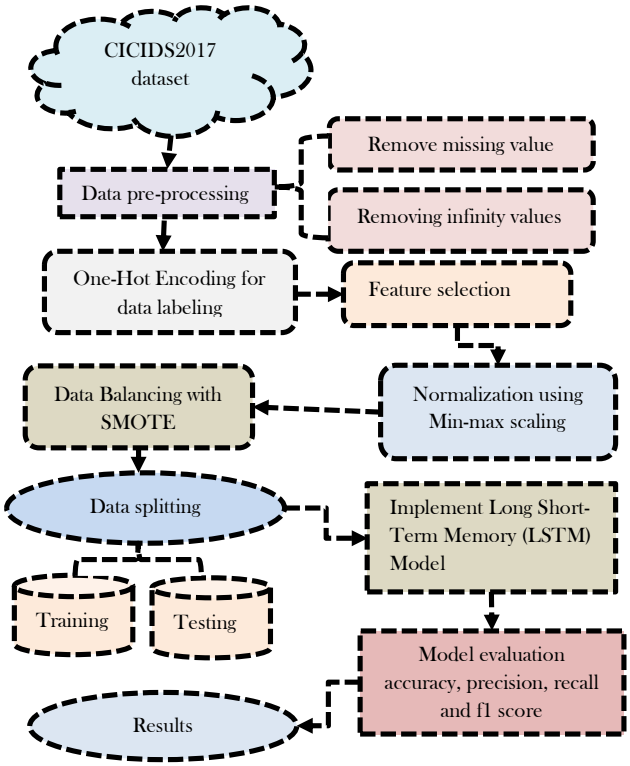

Fig 1: Proposed Flowchart for Threat Detection in Cloud Databases

3. Research Methodology

The methodology begins with pre-process the CICIDS2017 dataset involving removal of missing and infinite values, one-hot encoding for categorical labeling, feature selection, scaling. Split the cleaned data into training and testing sets after using SMOTE to fix the severe class imbalance. Finally, to ensure reliable Cloud Database Threat Identification, introduce an LSTM model for classification and assess it using industry-standard metrics such as F1, REC, ACC, and PRE. In Fig. 1 shows the whole procedure.

This section lays out the suggested workflow for cloud database threat detection in great detail.

3.1. Data collection

An updated set of data called the CICIDS2017 dataset is used to test the hypotheses in this research. Along with both harmless and harmful assaults, this collection has 28,30,743 records. Fourteen of the most common forms of cyberattacks are included in this list: DDoS, brute-force SSH, brute-force FTP, brute-force heartbleeds, and botnets. The following are examples of data visualisations that were employed to analyse the distribution of attacks, feature correlations, etc.: bar plots and heatmaps:

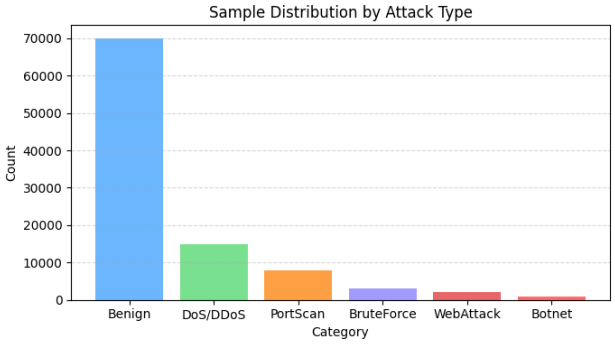

Fig 2: Count Plot for Class Distribution

Fig. 2 illustrates the distribution of samples within the dataset, highlighting the benign traffic compared to attack categories. The largest portion, shown in blue, represents benign samples, which make up the majority of the dataset. Conversely, DoS/DDoS, PortScan, BruteForce, WebAttack and Botnet types of attacks have much lower percentages which implies imbalance in classes. Among them, the proportion of DoS/DDoS and PortScan is much larger than the proportion of the other attacks, whereas WebAttack and Botnet are seen as significant fractions. This imbalance makes it clear that data balancing methods, including SMOTE, are necessary to provide equitable representation of each of the classes in the model training.

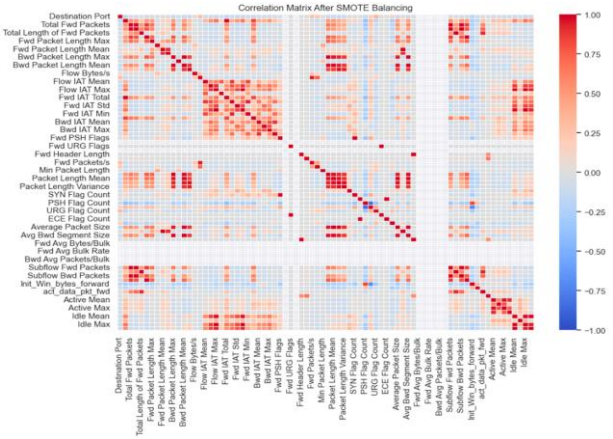

Fig 3: Correlation Heatmap of Features

Fig. 3 shows the connections between the features of the current dataset. When the correlation coefficient between two variables is between -1 (very negative correlation) and +1 (very positive correlation), the heatmap's cells display the corresponding correlation coefficients; a blue-red colour gradient denotes the direction and strength of the relationships. The diagonal values are perfect correlation (1.0) because each of the features is compared with itself. The matrix indicates that features that can be highly correlated with each other are considered to be potentially redundant, and those that have weak or no correlation. This kind of analysis is vital to the selection of features and dimensionality reduction because it could be used to filter out features that are too much correlated in order to contribute new value to the model.

3.2. Data pre-processing

Data preparation has been done based on the CICIDS2017 data set, which included concatenation, cleansing, and feature extraction. The dataset was further pre-treated by removing missing values, infinite values and noise. Also, the normalization and data transformation were used. The processing before processing is described as follows:

- Remove missing value: removing missing values should not compromise the dataset's overall dependability and quality. As a verification tool, it used the null().sum() method that ensured that the dataset was free of any missing values entirely after clean up.
- Removing infinity values: Investigated the dataset to determine if the infinity values were positive (inf) or negative (-inf) so that could identify the rows containing such values. The dataset was cleansed of any rows containing these values to ensure accurate model training and to avoid any potential distortions.

3.3. One-Hot Encoding for data labeling

Data labeling or data annotation refers to the process of tagging or identifying meaningful labels to raw data in order to render it useful to machine learning (ML) models. One-Hot Encoding is a data preprocessing method that transforms the data in a categorical form into a numerical form that the machine learning algorithms can comprehend. This is ideal especially when the categorical variables are nominal in nature in that the categories do not have an order.

3.4. Feature selection

Feature selection is the procedure by which a subset of the features present in a dataset that are deemed relevant are chosen from the original set of features. Eliminating superfluous or unneeded characteristics is a crucial part of machine learning that should enhance the model's performance, simplify it, and make it more understandable. The use of a feature selection algorithm can be viewed as the integration of a search method to propose new feature subsets, and a quality measure that ranks the subsets. The most straightforward algorithm is to search over all subsets of features that identify the subset that minimizes the error rate.

3.5. Normalization using Min-max scaling

Records normalization was done through min max technique to ensure that the values fell within a range of 0 to 1. This was done in a bid to maximize the performance of the classifiers in use and reduce the impact of outliers. Normalization was done based on the following mathematical formula (1):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (1)$$

where X initially represents the feature's value, X'its normalised value, X_{min} its minimum value, and X_{max} its maximum value.

3.6. Data Balancing with SMOTE

Data balancing is a method for addressing class imbalance in datasets by managing the distribution of classes. When one class is significantly under-represented in a classification task, data balancing is often used to improve the ACC of the model. One of the techniques that help in correcting imbalance of classes in datasets especially when training machine learning models is data balancing with SMOTE. Fig. 4 indicates the impact of SMOTE since the dataset is balanced in the synthetically generated samples of the minority classes, and the classes have an equal representation. This tuning increases the trustworthiness of the training procedure and increases the total output of machine learning models.

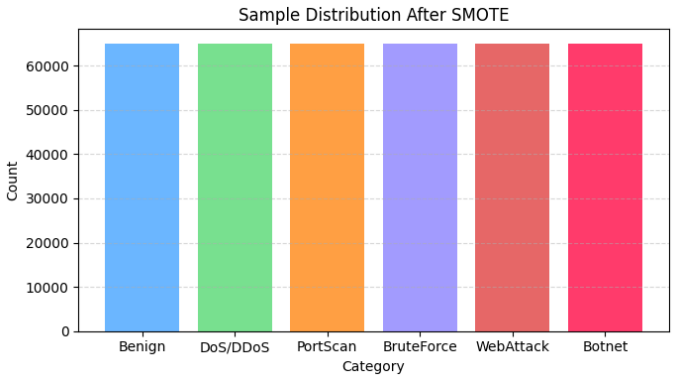

Fig 4: Class Distribution After Applying SMOTE

3.7. Data Splitting

The dataset was divided into training and testing sets as part of the data preparation process. To be more specific, 80% of the data set was used as training data, and 20% was used as testing data to evaluate the model's performance.

3.8. Proposed Long Short-Term Memory (LSTM) Model

The LSTM is great for classifying text because it can learn how words depend on each other over time. Layered support vector machines (LSTM) classifiers are a subset of RNNs, a type of layered network that feeds information about its outputs into its subsequent layers [26]. Because of its feedback connections, LSTM can process data in sequences rather than simply individual points. A long short-term memory (LSTM) node consists of a cell, an output gate, a forget gate, and an input gate. The three gates regulate the information flow within the cell, which is in charge of long-term value storage. Recurrently connected memory blocks with three multiplicative gates each make up long short-term memory (LSTM) layers. The gates in Equations (2) to (7) continuously read, write, and reset the temporary information to make sure it's used for the specified amount of time. The following updates are made to the unit's input, x_t, h_{t-1}, c_{t-1}, and the unit's output, h_t, c_t.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \qquad (2)$$

$$f_t = \sigma(W_f x_t + U_i h_{t-1} + b_f) \qquad (3)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \qquad (4)$$

$$g_t = tanh(W_g x_t + U_g h_{t-1} + b_g) \qquad (5)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \qquad (6)$$

$$h_t = o_t \odot \tanh(c_t) \qquad (7)$$

The logistic sigmoid function is represented by σ in the previous equations, whereas element-wise multiplication is denoted by $\odot$. The LSTM unit includes an input gate i_t, a forget gate f_t, an output gate o_t, a hidden unit $ht\,t$, and a memory cell $c\,c\,t\,t$ on each time step t. W and U are the parameters that were learnt, and there is an extra bias that is represented by b. To no one's surprise, the input gate controls the quantity of data written to each unit, the forget gate controls the quantity of data erased from the memory cell, and the output gate controls the quantity of data exposed from the internal memory state.

3.9. Evaluation metrics

The effectiveness of the proposed arrangement was evaluated using a number of performance indicators. By comparing the actual outcomes with the projected goals of the trained models, True Negatives (TN), False Negatives (FN), True Positives (TP), and False Positives (FP) were then computed. Equations (8) through (11) are then used to develop performance measurements including REC, ACC, PRE, and F1:

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+Fp+TN+FN}} \quad (8)$$

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \quad (9)$$

$$Recall = \frac{\text{TP}}{TP+FN} \quad (10)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (11)$$

The ACC of the trained model is the proportion of instances where it produced the correct forecast as a percentage of the total dataset examples. The ratio of the number of positive examples properly predicted to the total number of positive cases is the definition of the ACC of a model's predictions. REC is a measure of the ACC of positive event predictions relative to the total number of alternative outcomes. Finding a happy medium between the two is the goal of the F1, which is a combination of the harmonic mean of REC and PRE.

4. Results And Discussion

The provided method was replicated using a system that included 16 GB of random-access memory (RAM), a 250 GB solid-state drive (SSD), a 1 TB hard disc drive (HDD), an Intel Core i5-8600k processor, and a GeForce 1050Ti 4 GB. The suggested model was trained on the CICIDS2017 dataset and evaluated using various key performance metrics, including REC, ACC, PRE, and F1, as shown in Table II. The model was also very precise with the ACC of 98.51% showing that it is very apt in classification of both normal and malicious activities. The model has a very low false positive rate of 99 percent, and hence most of the threats identified are indeed malicious. These findings suggest that LSTM model is providing strong and stable threat detection capacity in cloud database systems.

Table 2: Experiment Results of Proposed Models for of Threat Detection in Cloud

Matrix	LSTM
Accuracy	98.51
Precision	99
Recall	98
F1-score	99

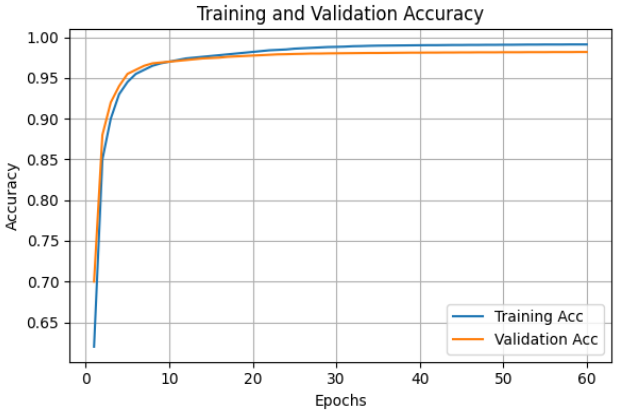

Fig 5: Training And Validation Accuracy Curves for The LSTM Model

The ACC of the proposed model during training and validation is displayed in Fig. 5, using 60 epochs. They learn well since the training ACC (blue line) is lower at the beginning but speeds up (and the validation ACC, orange line) as the epoch count increases. The two curves approach each other steadily and settle at over 95% as they exhibit steady performance with a slight variation as training advances. The fact that the ACC of the training and validation is close indicates that the model has a good generalization behavior and there is no evidence of overfitting that may illustrate that the model is robust and reliable in the detection of threats with ACC.

The training and validation loss of the proposed model, broken down by epoch, is shown in Fig. 6. The initial losses are also quite high and the training loss (blue) initially is close to 1.0 and the validation loss (orange) is close to 0.5. Nevertheless, the curves diminish drastically during the initial epochs, which demonstrates that the learning process is quick and the weight changes effectively. As the training advances, the losses start to decrease gradually and approach zero and there is only a few fluctuations in the validation loss which indicates that performance remains constant. The near correspondence between training and validation loss proves that the model does not overfit and has a high level of generalization, which is reliable and accurate detection of threats.

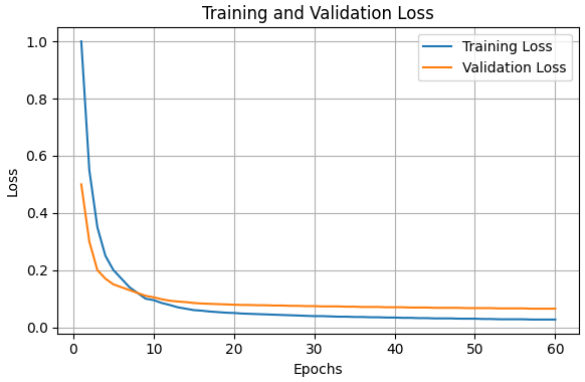

Fig 6: Training And Validation Loss Curves for The LSTM Model

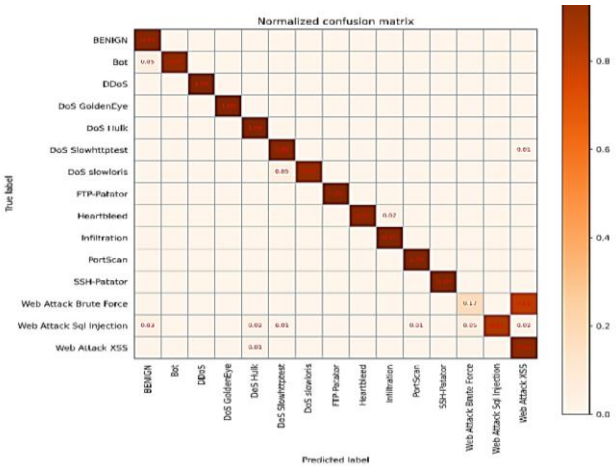

Fig 7. Plot The Confusion Matrix for The LSTM Model

The results of the proposed model classification on the various classes of attacks and benign traffic are shown in Fig. 7. The diagonal values, indicating the accurate classifications, are quite high, including 0.99 with BENIG, 0.99 with DoS GoldenEye, 0.99 with DoS Hulk, 0.98 with DoS Slowloris, 0.97 with FTP-Patator and 0.99 with PortScan, which means that the detection rates are excellent in these classes. Slightly smaller and yet high values are observed in a few classes, including 0.91 of Web Attack Brute Force, 0.92 of Web Attack SQL Injection and 0.91 of Web Attack XSS that represent misclassifications to a small extent. The small off-diagonal values: 0.05 and 0.12 show the small shares of cases, which were wrongly categorized into the other types of attacks. All in all, the matrix shows that the model has very high ACC in the detection of both benign and attack traffic with most classes having high ACC of more than 95% correct classification which proves that the model is effective and reliable in the detection of threats.

4.1. Comparative analysis

The efficiency of the proposed LSTM model is validated by comparing it to other existing models in Table III. Along these lines, MLP and SVM models have shown respectable detection rates of 96.4 and 96.0 %, respectively. XGBoost has a little worse yet, good performance and the ACC is 94.1% and the F1 is always at 93.7. Comparatively, DeepGFL achieves very lower levels of effectiveness, only 53.1% ACC and poor REC, which is one of the signs that it has a problem with generalization or feature learning. Overall, the findings indicate that deep learning methods, particularly LSTM, are more effective than traditional models at detecting cloud threats due to their ability to capture complex temporal patterns.

Table 3: Comparison Of Different Predictive Models of Threat Detection in Cloud Databases

Models	Accuracy	Precision	Recall	F1– score
MLP[27]	96.4	97.0	96.6	97
DeepGFL	53.1	-	44.8	53.1
XGB[28]	94.1	93.2	94.4	93.7
SVM[29]	96.0	93.12	99.27	96.10
LSTM	98.51	99	98	99

The ability to retain long-term dependencies and sequential patterns in network traffic, which traditional models do not address effectively, is a key benefit of the proposed LSTM model. Its gating mechanisms and memory cell facilitate it to determine more intricate behaviors and subtle anomalies, resulting in improved threat detection of cloud databases. The LSTM model had the top ACC as illustrated in the experimental results. This high ACC as well as its high PRE and REC are indicative of the reliability and strength of the LSTM model in reducing false alarms and guaranteeing effective cloud database security.

5. Conclusion And Future Study

This research looks at how deep learning-based IDSs can be used in large-scale network settings to deal with cyber risks that are getting smarter and more network traffic. Ineffective threat detection is a result of traditional IDS's complicated feature engineering, class imbalance in datasets, and high FPR. Using a combination of deep learning models can help overcome these limitations. The evaluation results show that when using the CICIDS2017 data to improve the ACC and efficacy of threat detection models in cloud databases, features selection is absolutely crucial. The models were able to remove unnecessary and unimportant features thus the models were able to concentrate on the most important attributes thereby making complex models simple and enhancing predictive ACC. The XGBoost 94.1%, MLP 96.4%, SVM 96%, and the proposed LSTM model were the highest performing except at 98.51%. The proposed LSTM model is very effective in the threat detection of cloud databases, but there are certain limitations. The study relies on a single dataset (CICIDS2017), and it may not be sufficient to capture evolving trends of attacks or other cloud traffic real-life practices.

Moreover, as SMOTE facilitates the balancing of classes, synthetic oversampling can also be biased in rare categories of attacks. Further research in this area can be based on proving the model on a variety of, heterogeneous data, the use of more complex imbalance managing strategies, and the investigation of hybrid deep learning models, which might combine LSTM with attention models or graph-based ones, and increase the detectability and scalability of various dynamic clouds.

References

[1] M. R. R. Deva, "Advancing Industry 4.0 with Cloud-Integrated Cyber-Physical Systems for Optimizing Remote Additive Manufacturing Landscape," in 2025 IEEE North-East India International Energy Conversion Conference and Exhibition (NE-IECCE), 2025, pp. 1–6. doi: 10.1109/NE-IECCE64154.2025.11182940.

[2] S. Thangavel, "AI Enhanced Image Processing System For Cyber Security Threat Analysis," 2024.

[3] A. R. Bilipelli, "AI-Driven Intrusion Detection Systems for Large- Scale Cybersecurity Networks Data Analysis : A Comparative Study," TIJER – Int. Res. J., vol. 11, no. 12, pp. 922–928, 2024.

[4] P. Nutalapati, J. R. Vummadi, S. Dodda, and N. Kamuni, "Advancing Network Intrusion Detection: A Comparative Study of Clustering and Classification on NSL-KDD Data," in 2025 International Conference on Data Science and Its Applications, ICoDSA 2025, 2025, pp. 880–885. doi: 10.1109/ICoDSA67155.2025.11157595.

[5] G. Sarraf, "Behavioral Analytics for Continuous Insider Threat Detection in Zero-Trust Architectures," Int. J. Res. Anal. Rev., vol. 8, no. 4, pp. 596–602, 2021.

[6] S. Narang and V. G. Kolla, "Next-Generation Cloud Security: A Review of the Constraints and Strategies in Serverless Computing," Int. J. Res. Anal. Rev., vol. 12, no. 3, pp. 1–7, 2025, doi: 10.56975/ijrar.v12i3.319048.

[7] M. K. Shah, "AI-Based Framework for Ransomware Detection in Android Systems: Enhancing Mobile Security," in 2025 5th International Conference on Artificial Intelligence and Signal Processing (AISP), IEEE, Nov. 2025, pp. 1–8. doi: 10.1109/AISP68263.2025.11396254.

[8] B. Madupati, M. M. Mohammed, L. Upadhyay, D. P. Guda, K. Kaushik, and M. Soni, "Integrating Artificial Intelligence with Cybersecurity for Resilient Wireless Communication Against Advanced Threats," in 2025 International Conference on Artificial Intelligence and Machine Vision (AIMV), IEEE, Aug. 2025, pp. 1–5. doi: 10.1109/AIMV66517.2025.11203666.

[9] S. Amrale, "A Novel Generative AI-Based Approach for Robust Anomaly Identification in HighDimensional Dataset," Int. J. Adv. Res. Sci. Commun. Technol., vol. 4, no. 2, 2024.

[10] A. Syed, "Securing IoT-Driven Supply Chains," in Supply Chain Software Security, Berkeley, CA: Apress, 2024, pp. 289–342. doi: 10.1007/979-8-8688-0799-2_7.

[11] B. R. Ande, "Autonomous AI Agents for Identity Governance: Enhancing Financial Security Through Intelligent Insider Threat Detection and Compliance Enforcement," Int. Conf. Data Sci. Big Data Anal., pp. 491–502, 2025.

[12] H. Ravilla, J. Yarra, and S. Dilip, "Role of SOQL and Database Optimization in Large-Scale Salesforce Implementations," Int. J. Eng. Archit., vol. 3, no. 1, pp. 13–31, Feb. 2026, doi: 10.58425/ijea.v3i1.481.

[13] V. Prajapati, "Enhancing Threat Intelligence and Cyber Defense through Big Data Analytics: A Review Study," J. Glob. Res. Math. Arch., vol. 12, no. 4, pp. 1–10, 2025.

[14] S. K. Chintagunta and S. Amrale, "Enhancing Cloud Database Security Through Intelligent Threat Detection and Risk Mitigation," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 8, no. 3, pp. 1–13, Dec. 2022.

[15] R. Dattangire, R. Vaidya, D. Biradar, and A. Joon, "Exploring the Tangible Impact of Artificial Intelligence and Machine Learning: Bridging the Gap between Hype and Reality," in 2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET), IEEE, Aug. 2024, pp. 1–6. doi: 10.1109/ACET61898.2024.10730334.

[16] H. B. Dama, "A Survey of MySQL Database Administration Techniques and Best Practices," ESP J. Eng. Technol. Adv., vol. 6, no. 1, pp. 89–98, 2026.

[17] S. B. Shah, B. Boddu, N. Prajapati, and S. A. Pahune, "AI-Powered Advanced Intrusion Detection for Securing Cloud Environments Against Network Attacks," in 2025 Global Conference in Emerging Technology (GINOTECH), IEEE, May 2025, pp. 1–7. doi: 10.1109/GINOTECH63460.2025.11076673.

[18] H. P. Cyril, "DeepNetDetect: A Deep Learning-Based Approach for Early Anomaly Detection in Network Traffic," in 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), IEEE, Feb. 2026, pp. 1–6. doi: 10.1109/ICAIC67076.2026.11395734.

[19] V. Verma, "Big Data and Cloud Databases Revolutionizing Business Intelligence," TIJER – Int. Res. J., vol. 9, no. 1, pp. 48–58, 2022.

[20] M. Dhinakaran, M. Sundhari, S. Ambika, V. Balaji, and R. T. Rajasekaran, "Advanced Machine Learning Techniques for Enhancing Data Security in Cloud Computing Systems," in 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), 2024, pp. 1598–1602. doi: 10.1109/IC2PCT60090.2024.10486559.

[21] G. Tiwari and R. Jain, "Detecting and Classifying Incoming Traffic in a Secure Cloud Computing Environment Using Machine Learning and Deep Learning System," in Proceedings - 2022 IEEE 7th International Conference on Smart Cloud, SmartCloud 2022, 2022. doi: 10.1109/SmartCloud55982.2022.00010.

[22] U. Garg, H. Sivaraman, A. Bamola, and P. Kumari, "To Evaluate and Analyze the Performance of Anomaly Detection in Cloud of Things," in 2022 13th International Conference on Computing Communication and Networking Technologies, ICCCNT 2022, 2022. doi: 10.1109/ICCCNT54827.2022.9984316.

[23] P. Ntambu and S. A. Adeshina, "Machine Learning-Based Anomalies Detection in Cloud Virtual Machine Resource Usage," in 2021 1st International Conference on Multidisciplinary Engineering and Applied Science, ICMEAS 2021, 2021. doi: 10.1109/ICMEAS52683.2021.9692308.

[24] T. L. Yasarathna and L. Munasinghe, "Anomaly detection in cloud network data," in 2020 International Research Conference on Smart Computing and Systems Engineering (SCSE), IEEE, Sep. 2020, pp. 62–67. doi: 10.1109/SCSE49731.2020.9313014.

[25] T. Salman, D. Bhamare, A. Erbad, R. Jain, and M. Samaka, "Machine Learning for Anomaly Detection and

Categorization in Multi-Cloud Environments," in 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), 2017, pp. 97–103. doi: 10.1109/CSCloud.2017.15.

[26] R. Patel, "Automated Threat Detection and Risk Mitigation for ICS (Industrial Control Systems) Employing Deep Learning in Cybersecurity Defence," Int. J. Curr. Eng. Technol., vol. 13, no. 06, pp. 584–591, Dec. 2023, doi: 10.14741/ijcet/v.13.6.11.

[27] S. N. Pakanzad and H. Monkaresi, "Providing a hybrid approach for detecting malicious traffic on the computer networks using convolutional neural networks," in 2020 28th Iranian Conference on Electrical Engineering, ICEE 2020, 2020, pp. 1–6. doi: 10.1109/ICEE50131.2020.9260686.

[28] G. Nassreddine, M. Nassereddine, and O. Al-Khatib, "Ensemble Learning for Network Intrusion Detection Based on Correlation and Embedded Feature Selection Techniques," Computers, vol. 14, no. 3, pp. 82–104, 2025, doi: 10.3390/computers14030082.

[29] T.-H. Chua and I. Salam, "Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection Using Progressive Dataset," Symmetry (Basel)., vol. 15, no. 6, p. 1251, Jun. 2023, doi: 10.3390/sym15061251.