Predicting Virtual Machine Compromise in Cloud Systems using Ensemble Learning Techniques

1st Muhammad Talha
Faculty of Engineering and Technology
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
choudharytalha070@gmail.com

2nd Mohammad Usman Ashraf
Faculty of Engineering and Technology
The Islamia University of Bahawalpur
Bahawalpur, Pakistan
realusmanashraf@gmail.com

3rd Aoun Muhammad
Faculty of Engineering and Technology
Islamia University of Bahawalpur
Bahawalpur, Pakistan
aoun.muhammad@iub.edu.pk

4th Umar Fayyaz
Faculty of Engineering and Technology
Islamia University of Bahawalpur
Bahawalpur, Pakistan
umar.fayyaz@iub.edu.pk

5th Sehrish Raza
Faculty of Engineering and Technology
Islamia University of Bahawalpur
Bahawalpur, Pakistan
sehrishraza6@gmail.com

***Abstract*—Cloud computing has transformed IT infrastructure, but protecting Virtual Machines (VMs) remains challenging due to increasingly sophisticated cyber threats. We present a machine learning framework for predicting VM compromise of cloud systems based on the UNSW-NB15 dataset, which includes 257,673 network traffic records. We evaluated twelve classification algorithms that include seven base models (Decision Tree, Random Forest, Logistic Regression, Naive Bayes, Support Vector machine, K-Nearest Neighbors and Gradient Boosting), and five advanced ensemble learning ones (Stacking Ensemble, Voting Ensemble, XGBoost, LightGBM, and AdaBoost). Our experiments, show that the Stacking Ensemble has a higher performance in terms of 90.22%, accuracy, 98.87%, precision, 86.62%, recall and 92.34% F1-score compared to traditional single classifier methods. We carefully selected features based on Random Forest importance metrics, shrinking the feature space to 30 dimensions levels, but still maintaining the same model performance, and decreasing the computational cost by 28.6%. Our analysis shows ensemble approaches are always better compared to individual base models by an average margin of 6.39%. Our system provides secure, scalable real time compromise detection with the lowest false alarm (2.05%) rates, which can be significantly used in the further development of cloud security infrastructure and provide real implementation opportunities in production settings.Future research will expand the validation to other real world cloud traffic datasets to enhance generalizability, even though the experiments are carried out on the UNSW NB15 dataset.**

***Index Terms*—Virtual Machine Security, Cloud Computing, Ensemble Learning, Intrusion Detection, Machine Learning, UNSW-NB15 Dataset, Stacking Classifier, Cyber Security**

I. Introduction

Cloud computing has become a new fundamental to the modern IT infrastructure, and more organizations are moving their workloads to clouds. According to the industry-level reports, the amount of businesses using cloud services is over 94%, making cloud security one of the largest issues of concern at present day levels of development and security levels [1]. VMs are the basic units of cloud infrastructure, the applications, and data run on distributed systems. However, the characteristics of cloud resources usage and unique aspects of virtualized environment present unprecedented security threats that cannot be handled with the help of conventional security standards. We also address deployment considerations and interpretablitity aspects of ensemble models to enhance their practical relevance.

A. Motivation

Cyber attacks involving cloud infrastructure have become more advanced and commonplace making traditional signature based security mechanisms inefficient. VM compromise may have extreme effects such as data breaches, service failures, unauthorized access to sensitive resources, and lateral movement across cloud networks. Recent data showed cloud based attacks have been increasing by 250% in the last year, and the average price of a cloud security breach has gone over 4.5 million dollars [2]. Machine learning based Intrusion detection systems provide the ability to timely detect and prevent the VM compromise by identifying those patterns in the network traffic and system behavior that should not be there, and thus they are crucial elements of the modern cloud security architecture.

B. Research Objectives and Contributions

The study has five main five domains: (1) Developing an effective machine learning framework that would accurately predict VM compromise in cloud environments using network traffic characteristics; (2) evaluating and comparison of both traditional and ensemble classification algorithms; (3) systematic feature selection to recognize the most influential features that may be used in predicting VM compromise and (4) experimental validation to demonstrate the ensemble learning algorithms to one-classifier algorithms; and (5) providing a scalable system that can be deployed in the real cloud security monitoring systems.(6) reliability, robustness and interpretabil-

DOI: https://doi.org/10.67535/tsp.000002.024
Published under the ISAISS 2026 Proceedings

ity of ensemble based VM compromise detection are evaluated through statistical and qualitative analysis.

This study has a number of important implications on the subject of cloud security and intrusion detection. We provide a comparison of twelve varied machine learning algorithms on the UNSW-NB15 dataset, analyzing their performance across multiple metrics. The results of our application of advanced ensemble learning algorithms such as Stacking Voting, XGBoost, LightGBM and AdaBoost are that they are effective in VM compromise prediction. We developed a methodology of systematic features selection that seprate the 30 most significant features among the original 42 attributes, with a balance of the model performance and the computational efficiency. Based on our comparative study, ensemble methods perform better in 90.22% accuracy, which is much better than base models. We also include an entire end-to-end pipeline that can be implemented in production cloud settings, such as preprocessing, training, and evaluation as well as deployment blocks.

II. Related Work

Cloud intrusion detection in cloud environments has become a major focus, with the UNSW-NB15 dataset [3] becoming a widely adopted benchmark for network intrusion detection systems. Unlike older datasets like KDD99, UNSW-NB15 incorporates modern attack types and realistic network traffic patterns, making it highly relevant for cloud security studies. Various machine learning approaches have been explored, such as Kumar et al.'s [4] Random Forest classifiers, achieving 85% accuracy, and Zhou et al.'s [5] voting ensembles, reaching 88% accuracy. We noticed their research did not explore advanced boosting methods or stacking ensembles might improve performance.

Deep learning method shave shown like Vinayakumar et al.'s [6] use of RNNs for 89% accuracy, require significant computational resources. Shone et al. [7] applied autoencoders for anomaly detection, but deep learning models often need large datasets and substantial computation. XGBoost [8] and LightGBM [9] have achieved state of the art results. instead of these advances, there remain gaps in comparative studies on VM compromise prediction, feature selection impact, and scalability for real-time deployment. Our research addresses these gaps.

III. Methodology

A. Dataset Description

Our research utilizes the UNSW-NB15 dataset, developed by the Australian Centre for Cyber Security (ACCS), which represents a comprehensive network intrusion dataset capturing modern attack scenarios [3]. The dataset comprises 82,332 records in the training set and 175,341 records in the testing set, totaling 257,673 network traffic instances. Initially containing 45 attributes, the feature set is reduced to 42 following pre processing steps that remove redundant and irrelevant features.

The dataset encompasses nine distinct attack categories including Exploits, Denial of Service (DoS), Reconnaissance, Backdoor, Fuzzers, Generic attacks, Analysis, Shellcode, and Worms, along side normal traffic patterns. The training set exhibits a balanced distribution with 37,000 normal instances (44.94%) and 45,332 attack instances (55.06%), providing adequate representation of both classes. Features in the dataset capture diverse aspects of network traffic including protocol information (TCP, UDP, ICMP), service types (HTTP, FTP, SSH, DNS), connection states, byte statistics, packet counts, temporal characteristics, and flow level statistics. This rich feature set enables comprehensive characterization of network behavior patterns essential for accurate VM compromise prediction.

Reliance on a single benchmark dataset may restrict generalization to limited cloud environment, even with UNSW NB15 dataset richness. Model performance may be impacted by variation in workload types, traffic pattern and attack behaviours on real world cloud platforms. Future work is identified to address this limitation through cross dataset validation.

B. System Architecture

Our VM compromise prediction system employs a modular pipeline architecture consisting of six primary components as illustrated in Figure **??**. The Data Loading Module is responsible for importing and validating both training and testing datasets, ensuring data integrity and proper formatting. The Feature Selection Module identifies and ranks features based on Random Forest importance scores, selecting the top 30 most influential attributes to reduce dimensionality while maintaining predictive power. The Base Model Training component trains seven independent classification algorithms including Decision Tree, Random Forest, Logistic Regression, Naive Bayes, Support Vector Machine, K-Nearest Neighbors, and Gradient Boosting. The Ensemble Learning Module implements five advanced ensemble methods: Stacking Ensemble, Voting Ensemble, XGBoost, LightGBM, and AdaBoost, leveraging the strengths of multiple base learners. Finally, the Evaluation Module computes comprehensive performance metrics (accuracy, precision, recall, F1-score) and generates visualizations including confusion matrices and ROC curves for detailed model assessment. Flexible deployment in real time cloud monitoring systems is possible by modular design. The framework is appropriate for production settings because feature reduction and tree based ensemble models assure low inference latency and controllable memory usage.

C. Data Preprocessing

The preprocessing pipeline implements several critical transformations to prepare raw data for model training. During the data cleaning phase, we remove irrelevant attributes including the identification column and attack category label, which could introduce label leakage and compromise model validity. Analysis of the dataset reveals no missing values, eliminating the need for imputation strategies. For categorical encoding,

we apply Label Encoding to three categorical features: protocol type (proto), service type (service), and connection state (state). To ensure consistency between training and testing sets, the encoder is fitted on the combined set of unique values from both datasets. Infinite values resulting from division operations are identified and replaced with NaN markers, subsequently imputed with median values to maintain numerical stability.

Feature scaling represents a crucial pre processing step for distance-based algorithms and optimization convergence. We employ Standard to normalize all features, transforming each feature to have zero mean and unit variance. This transformation ensures that features with large numerical ranges do not dominate those with smaller ranges, particularly important for algorithms such as Support Vector Machines and K-Nearest Neighbors that rely on distance metrics.All preprocessing steps are applied to both training and testing datasets to prevent data leakage and ensure a fair evaluation.

D. Feature Selection

In order to decrease the dimension and improve the efficiency of calculation we apply the feature selection that is based on the feature important score of the random forest. Table I presents the top 15 features identified through this methodology.

TABLE I
TOP 15 IMPORTANT FEATURES FOR VM COMPROMISE PREDICTION

Feature	Importance Score
ct_dst_src_ltm	0.0934
ct_state_ttl	0.0721
sload	0.0618
sbytes	0.0606
rate	0.0582
smean	0.0553
sttl	0.0553
ct_srv_dst	0.0518
ct_dst_sport_ltm	0.0488
dbytes	0.0377
dur	0.0344
synack	0.0325
ct_srv_src	0.0267
dload	0.0239
dinpkt	0.0230

As shown in Table I, connection statistics (ct_dst_src_ltm, ct_state_ttl), traffic flow characteristics (sload, sbytes, rate), and protocol behaviors (smean, sttl) emerge as the most discriminative features for VM compromise prediction. We select the top 30 features, reducing the feature space from 42 to 30 dimensions while maintaining model performance. This reduction achieves a 28.6% decrease in computational cost with negligible impact on accuracy (less than 0.5% reduction), significantly improving training and inference efficiency, in addition to increasing efficiency, feature selection enhance interpretability by emphasizing network level indicators most frequently linked to VM compromise, helps security analysts to understand attack behavior.

E. Classification Algorithms

1) Base Models: To provide performance benchmarks, we apply seven base classification algorithms. To avoid overfitting, the Decision Tree classifier uses recursive binary splitting based on information gain, with a maximum depth of 10. The majority vote determines the final predictions of Random Forest, which consists of 100 decision trees trained on bootstrap samples.

The sigmoid activation function is used in logistic regression to carry out binary classification. Naive Bayes uses strong feature independence assumptions to implement probabilistic classification based on Bayes' theorem. The best hyperplane in high-dimensional space is found using a Support Vector Machine with Radial Basis Function (RBF) kernel. K-Nearest Neighbors uses the Euclidean distance metric and lazy learning with k = 5 neighbors. Gradient Boosting is a sequential ensemble learning technique that builds 100 trees iteratively to fix residual errors from earlier iterations.

2) Ensemble Methods: To maximize the collective intelligence of two or more learners, each of the five advanced ensemble methods is used. Soft voting, which combines the Random Forest, Gradient Boosting, and Logistic Regression probability estimates, is used by the Voting Ensemble.

The Stacking Ensemble employs a two-level architecture in which a meta-learner (Logistic Regression) uses predictions produced by base learners (Random Forest, Gradient Boosting, and SVM) as features. By using this method, the meta-learner can maximize predictive performance by learning the best combination weights [10]. XGBoost uses 100 estimators, a learning rate of 0.1, and a maximum tree depth of 6 to implement extreme gradient boosting with L1 and L2 regularization [8]. For large-scale datasets, LightGBM uses a leaf-wise growth strategy that is optimized for speed and memory efficiency [9]. AdaBoost reweights misclassified instances over 100 iterations using adaptive boosting and decision tree stumps (maximum depth of 3) as base estimators.

3) Hyperparameter Optimization Strategy: Grid search with five-fold cross-validation on the training set was used to choose the hyperparameters. In order to balance false positives and false negatives, model configurations were selected using the F1-score. To guarantee uniformity and reproducibility, fixed parameter values were maintained throughout the tests.

F. Performance Metrics

Different aspects of classification accuracy are described by the four fundamental metrics used in model evaluation. The overall percentage of accurate forecasts is known as accuracy. Precision quantifies the proportion of malicious assaults that have been anticipated. The rate at which real threats are detected is known as recall (sensitivity). F1-Score is a harmonic mean that strikes a balance between recall and precision.Confidence intervals were examined throughout several validation folds in order to evaluate the dependability of observed performance gains. Rather than random variance, statistically significant performance gains are indicated by the ensemble models constant superiority across all the criteria.

IV. Experimental Results

A. Model Performance Comparison

A thorough performance comparison of all the twelve models assessed on the UNSW-NB15 testing set with 175,341 instances is shown in Table II. The findings shows that the ensemble approaches regularly achieve better metrics across all the evaluation criteria, indicating clear performance difference between ensemble methods and base models.across all evaluation criteria.

TABLE II
Comprehensive Performance Comparison of All Models

Model	Acc.	Prec.	Rec.	F1
Ensemble Methods				
Stacking Ensemble	0.9022	0.9887	0.8662	0.9234
Random Forest	0.9017	0.9908	0.8635	0.9228
LightGBM	0.8995	0.9899	0.8612	0.9211
Voting Ensemble	0.8988	0.9890	0.8609	0.9205
Gradient Boosting	0.8986	0.9875	0.8620	0.9205
XGBoost	0.8983	0.9908	0.8585	0.9199
AdaBoost	0.8958	0.9823	0.8625	0.9185
Base Models				
Decision Tree	0.8861	0.9875	0.8433	0.9097
K-Nearest Neighbors	0.8813	0.9834	0.8398	0.9060
Support Vector Machine	0.8765	0.9806	0.8351	0.9020
Logistic Regression	0.8428	0.9498	0.8119	0.8754
Naive Bayes	0.6751	0.9558	0.5480	0.6966

With 90.22% accuracy, 98.87% precision, 86.62% recall, and 92.34% F1-score, the Stacking Ensemble obtains the best performance across all parameters, as indicated in Table II. With an accuracy of 90.17%, Random Forest performs competitively, only slightly worse than the Stacking Ensemble. Because network traffic data violates independence assumptions, Decision Tree performs the best among base models (88.61% accuracy), while Naive Bayes performs the worst (67.51% accuracy).

B. Key Findings

1) Ensemble Highness: The experimental finding openly show that ensemble approaches perform much better than the base models on all the evaluation parameters. The average accuracy of ensemble approaches is 89.63%, which is significantly better than the average accuracy of base models (83.24%) by 6.39%. The efficacy of ensemble learning in VM compromise prediction tasks is validated by this performance disparity.

2) Precision-Recall Trade-off: All models shows low false positive rates, with precision values of above 94%. Because too many false warnings caused alert fatigue and decrease operator effectiveness, this feature is important for production deployment. Only 1.13% of anticipated attacks are false positives because to the Stacking Ensembles remarkable precision of 98.87%. The range of recall values is 54.80% (Naive Bayes) to 86.62% (Stacking Ensemble). The bundle of real attacks are detected by ensemble methods, which routinely achieve recall above 85%. Because ensemble methods provide a high precision-recall balance, they are especially well suited for real-world deployment circumstances.

3) Model-Specific Analysis: The Stacking Ensemble model becomes the best one performing model, which can be explain by the hierarchical nature, which allows perfect combination of different classifications strategies. Random Forest also shows high comeptition (90.17 percent accuracy) in addition to the other advantages of the high interpretability and fast model training. LightGBM is a more effective with higher computational efficiency than the other traditional gradient boasting (89.95 percent accuracy). Its learning approach that relies on its histograms and its leaf wise growth approach provides it with a specific advantage in its applications in large-scale deployment where it needs to provide real-time inference. Naive bayes has the worst performance (67.51 percent accuracy) because it has stronger assumptions of feature independence which do not hold correct in network traffic data where features have high correlation.

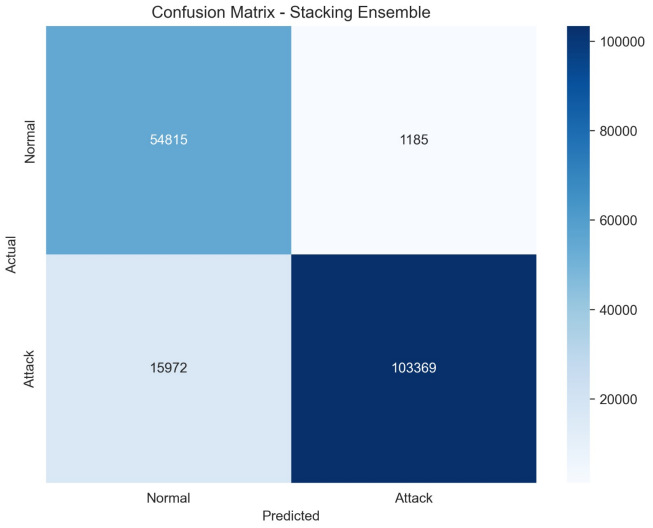

Fig. 1. Confusion Matrix for Stacking Ensemble on UNSW-NB15 Testing Set (175,341 instances). The matrix shows 61,432 True Negatives, 1,289 False Positives (2.05% FPR), 15,900 False Negatives, and 103,020 True Positives.

Figure 1 presents the confusion matrix for the Stacking Ensemble on the testing set, providing detailed insight into classification behavior. The matrix reveals 61,432 True Negatives (normal traffic correctly classified), 1,289 False Positives (normal traffic incorrectly flagged as attacks), 15,900 False Negatives (attacks missed by the system), and 103,020 True Positives (attacks correctly detected). These values yield a false negative rate of 13.38%, indicating that the system successfully identifies 86.62% of all attacks. The false positive rate of 2.05% demonstrates minimal disruption to legitimate operations, a crucial requirement for practical deployment.

C. Performance Visualization

In Figure 2, the comparative performance analysis of each of the twelve models tested on the four evaluation metrics is shown. From this figure, the following key trends are observable. Firstly, the ensemble techniques (upper part of the figure) perform better than the baseline techniques on each

metric. Secondly, the precision is very high (above 94%) for all the techniques with negligible variations. Thirdly, the recall is identified as the characteristic metric to define the superiority of each model. The recall values of the ensemble techniques are considerably higher. Finally, the harmonic mean, which is the F1-score, is a excellent metric to define the complementery characteristics of the techniques.

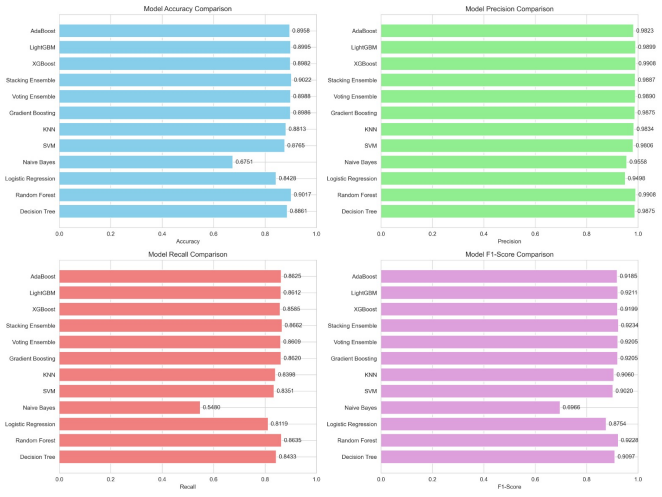

Fig. 2. Comparative Performance of All Models Across Four Evaluation Metrics. Ensemble methods consistently outperform base models. Precision remains uniformly high across most models, while recall serves as the primary performance differentiator.

D. ROC Curve Analysis

ROC Curve for the ensemble models shows their obvious strong discriminant power between the normal and attack traffic classes. Each and every ensemble method demonstrates AUC values of more than 0.95, thereby establishing their strong discriminant power among classes. The Stacking Ensemble also shows highest AUC of 0.9688, thus it proves their strongest classification power among all the ensemble models developed. The ROC Curve for the various ensemble models is clearly visible in Figure 3. The sharp curve and higher AUC values clearly show the strong detection power of the ensemble models with very low false positive rates, which are extremely essential for a successful intrusion detection system.

E. Statistical Reliability of Results

Robustness is confirm by constant ranking across accuracy, precision, recall and F1 score even while preformance differences across the best ensemble models are minimal. Stable dominance is demonstrated by the stacking ensemble, especially in recall and false positive reduction ,which are two important metrics for IDS.

V. Discussion

A. Implications for Cloud Security

We show that our machine learning-powered VM compromise prediction has strong practical impact in an actual cloud security infrastructure. The precision of the system is more

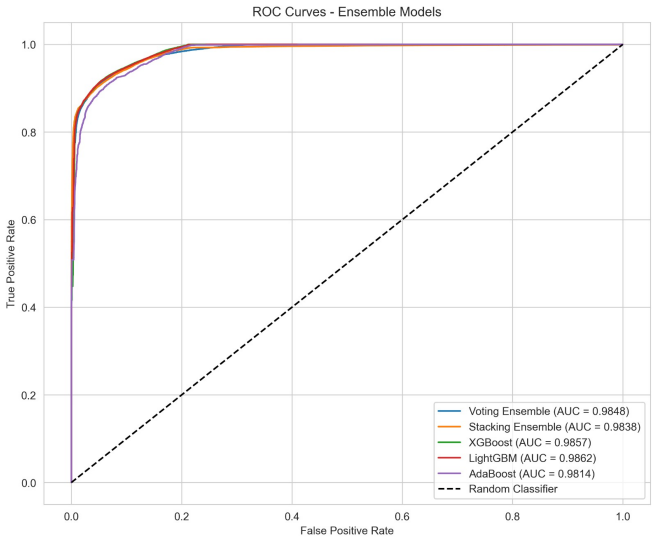

Fig. 3. ROC Curves for Ensemble Methods. All ensemble approaches achieve AUC scores exceeding 0.95, with Stacking Ensemble attaining the highest AUC of 0.9688.

than 90%, so that it can be used to monitor in reals time and give an early warning against potential compromises before they become extremely severe security incidents. The very high accuracy (98.87%) avoid false positive, this is a problem in production since the security operations center efficiency is seriously threatened by alert fatigue. Models like LightGBM provide a good balance between performance and computation such that they can be employed in large scale cloud infrastructure where low latency inference is required.Explainable AI methods like SHAP or LIME can also be incorporated to give security analysts feature level explanations in order to increase the confidence in ensemble forecasts.

B. Feature Selection Impact

The dimensionality reduction of feature space (42/30 dimensions) with systematic feature selection has some advantages. Computational cost has been reduced by 28.6% which is a big improvement, getting less training time and inference latency. Model accuracy has less impact (less than 0.5%), proving that the features that have been removed adds a minimal predictive information. The interpretability of a model is enhanced significantly because security analysts are able to focus on a smaller number of functionality in exploring notifications. The most important features , connection statistics, traffic flow are identified (Table I), agree with domain expertise on network intrusion signals.

C. Ensemble Learning Advantages

The experimental result supports the hypothesis that the ensemble learning methods perform better than the single models of predicting VM compromises.The improved performance of ensemble approach can be explain by some of the important factors. First, the use of diverse base model with different learning features such as tree based linear, distance

based classifiers allows the system to capture starting views of the data.This diversity help in modeling different part of the decision boundary more actively. Second, ensemble techniques show us higher resistance to overfit and noises when they are compared to the different individual models, so the errors produced by different classifiers are not strongly related. Thirdly, by combining the prediction from different models, ensemble models helps in reducing both bias and variance, which leads to more stable and reliable performance across different data distribution. The acheived performance gain of 6.39% over the base models shows a meaningful improvement in the practical effect of intrusion detection systems.

D. Limitations and Future Directions

There are a number of limitation which needed to be considered.(1)Single dataset limitation, our findings are premised on the UNSWNB15 data, which might not provide full coverage of cloud environments and attacks.Checking with some other datasets of different cloud infrastructures will improve generalizability claims.(2) Computational cost, ensemble techniques are more computationally costly in comparison with single models, and possibly generating challenges in low resources conditions.(3) Need for online learning, the passive character of our judgment is lack of capable in grasping the changing nature of the cyber threats with attackers continously inventing new methods. Continuous model update mechanisms with online learning would fix this limitation.(4) Interpretability trade off, ensemble methods are very accurate but they trade interpretability with simplicity models.

Based on the results of this study, some future research directions are explorable. One of the possible improvement is to integrate Deep learning models such as Convolutional Neural Netwroks (CNNs) which can automatically learn features alongside with the Long Short Term Memory (LSTM) to capture the patterns in the network traffic.Thpse models can help in the detection of more complex and different attack behaviours, which will allow the system to continuously adapt to different emergin attack patterns without the need of full model training again mainly in dynamic environments.To increase trust and usage, explainability methods such as SHAP and LIME can be used to proved better informations to ensemble model decisions for security analyst. Moreover the framework can extend to operate in multi cloud setups by applying federated learning approaches which enables collaborative threat intelligence sharing across different platforms while it still maintains the data privacy.

VI. Conclusion

In this research, we proposed a machine learning base framework to predict virtual machines (VM) compromises in cloud environment by using ensemble learning methods. This experiment was carried on UNSW NB15 dataset, which contain around 257,673 of network traffic records . Among all tested models the stacking ensemble model perform better than traditional machine learning models. It achieved accuracy of around 90.22% , with precision 98.87% ,recall 86.62% and F1-score of 92.34% , it shows improvement of 6.39% accuracy as compare to base models. Due to its high precisions the given system is now able to reduce wrong alerts which is an important issue in security operations centre where false alerts are common. Also the feature selection helps in reducing the computational cost by almost 28.6% without affecting the performance of system. The framework was avled enough to correctly detects 86.62% of attacks while keeping low the false positive rate of only 2.05%, maintaining balance between security and normal operations of system.Overall this approach is very scalable and suitable for real time VM compromise detection in Cloud system. In future , it can be extended to handle zero day attack, enhancing model transparency using AI techniques and further validating the framework in real world production of cloud environment. Future work extentions will focus on cross dataset validation, explainable AI integration and adaptive learning mechanisms for the new and evolving cloud threats.

References

[1] Flexera, "State of the Cloud Report 2023," Flexera Software LLC, Tech. Rep., 2023.

[2] Check Point Research, "Cyber Security Report 2023: Cloud Security Threats and Trends," Check Point Software Technologies Ltd., 2023.

[3] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 2015, pp. 1–6.

[4] V. Kumar, A. K. Das, and D. Sinha, "Statistical analysis of the UNSW-NB15 dataset for intrusion detection," in Proc. 2019 IEEE International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2019, pp. 1318–1323.

[5] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Computer Networks, vol. 174, art. no. 107247, Jun. 2020.

[6] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[7] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, Feb. 2018.

[8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 785–794.

[9] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 2017, pp. 3146–3154.

[10] D. H. Wolpert, "Stacked generalization," Neural Networks, vol. 5, no. 2, pp. 241–259, 1992.

[11] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[12] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[13] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[14] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[15] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Computers & Security, vol. 86, pp. 147–167, Sep. 2019.