National College of Ireland

Intrusion Detection System for Cloud ERP's using ML Techniques

Research Project
Msc Cloud
Computing

Vinay Kalidindi

Student ID:
x23107316

School of Computing
National College of
Ireland

Supervisor: Shivani Jaswal

National College of Ireland

Project Submission Sheet

Student Name: …… Vinay Kalidindi ……

Student ID: ……x23107316……

Programme: …Msc Cloud Computing…… **Year:** ……2024……

Module: ……Research Project……

Lecturer: ……Shivani Jaswal……

Submission Due Date: ……29/01/2025……

Project Title: Intrusion Detection System for Cloud ERP's using ML techniques

Word Count: ……6870……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: …Vinay Kalidindi……

Date: ……29/01/2025……

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on a computer. Please do not bind projects or place them in cover unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a failure.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Intrusion Detection System for Cloud ERP's using ML techniques.

Student Name: Vinay Kalidindi

Student ID: X23107316

Abstract

This paper argues that the evolution of ERP systems in cloud environments has increased the flexibility and availability of business processes. But it causes new risks for security problems: unauthorized access, data leaks, and cyber threats are favored by the nature of the cloud environment being open and shared. These make the integration of Intrusion Detection Systems (IDS) with cloud-based ERPs a crucial area of research owing to the arises of challenges. IDS are important for creating filters for monitoring the network traffic and timely detection presence of intruders to offer a protective layer for the network against intrusion. This project uses Azure Machine Learning Notebook for constituting the different ML models, such as Random Forest, Logistic Regression, and Support vector machine for building, training, and testing. The Random Forest classifier achieved an accuracy of 99.94%, precision of 99.97%, and recall of 99.88%, making it highly effective in distinguishing between normal and malicious network traffic. By integrating Microsoft Forms, Power Automate, and Azure Machine Learning, the system ensures real-time detection and automation for ERP-like scenarios. This study offers practical insights into improving ERP system security through scalable and adaptive ML solutions. The models were built based on the pre-processed KDD Cup 1999 data set which is considered to enclose all kinds of network activities and anomalous events. An application user interface was created using Stream lit which allowed the model to predict and monitor intrusion attempts in real-time. To mimic the functionality of an ERP system, several forms were created while Microsoft Power Automate was used to automate the selected workflows, which in their turn would generate emails upon intrusions. By combining these technologies, the IDS continuously tracks the ERP like activity, analyses inherent vulnerabilities, and informs the administrator regarding threats, making the IDS active in providing security. This project shows that it is possible to integrate machine learning and automation with cloud tools to improve ERP system security and create novel solutions based on them for enterprise environments. The recommendations drawn from this research are useful in enhancing basic protection and strengthening of ERP systems based on cloud.

Keywords: Intrusion Detection System (IDS), Machine Learning (ML), Cloud-based ERP, Random Forest, Anomaly Detection.

1. Introduction

The report is structured as follows: Section 2 provides a literature review of existing IDS methods and machine learning applications. Section 3 outlines the methodology, including data preprocessing and ML model selection. Section 4 covers design and implementation, highlighting the use of Microsoft Forms, Power Automate, and Azure ML. Section 5 discusses results, including model performance and analysis. Finally, Section 6 concludes the findings and suggests future work.

Background: Data and application processing has become one of the critical success factors for organizations and cloud computing has emerged as the enabling technology to support these business processes in recent years. Of these solutions ERP solutions have garnered much attention as they combine critical organization processes like financial, human resources, and supply chain into one system. It also makes it easy to manage data, optimize organizational operations while giving real-time data that organizations can use to make sound decisions. Consequently, organizations have migrated their ERP to the cloud with increased access and less expense on infrastructure requirements. However, the integration

of ERP systems into cloud environments brings significant security issues, although there are several benefits associated with this action. Traditional on-premises-Based ERPs, as they lack agility, are prone to cyberattacks because they are based on the internet, shared infrastructure, and difficult access management environment. Unauthorized access, data loss and compromises have severe threats towards the business-critical data stored in cloud servers. That is why sustaining strong protective layers is so crucial for organizations who depend on cloud-based ERP technology.

Importance: IDS have a critical function within this setting since they can observe traffic circulation within the network and distinguish activities that seem suspicious, likely to stem from a security breakthrough. IDS are intended for the discovery of specific and unspecific intrusions, informing administrators about an event in progress or a threat in time. Signature-based methods, rule- based methods are traditional IDS, and these often fail into the quickly evolving advancement of cyber threats, they cause high false-positive rates and missed incidents. Several challenges have hence been associated with traditional IDS, and the incorporation of machine learning (ML) algorithms as a solution to these challenges has therefore become apparent. That is why Machine Learning based IDS in comparison with other models can adapt to previously non-linear patterns and recognize new attacks, increase the DON'T know rate, and decrease the number of false positives. Different ML techniques, such as SVM, Decision Trees, Random Forest, and Neural Network, have been used to evaluate the efficiency of each type within intrusion detection. All the algorithms are different, and it each has its advantage for example, Decision Trees can generate easily interpretable results while at the same time Random Forests can be resistant to overfitting and Neural Networks can model patterns.

Therefore, ERP security in cloud environment maintains business operations and preserves data from secure threats. A lot can be done in terms of implementing higher ML algorithms in IDS that can work to provide an optimistic route in terms of having stronger security measures that are fit to follow the changing and uncertain space of cloud computing. The purpose of this work is to share evidence-based knowledge of the application of ML-based approaches to distinguish between genuine and malicious activities within cloud-environments hosting ERP applications and fill the gap between theory and practice. This research aims to address the following research questions:

How can machine learning techniques be effectively applied to develop intrusion detection systems for cloud-based ERP systems?

To answer this question, the following objectives are defined:

- Determine as many threats as possible and discover how some of these threats are unique to the cloud ERP environment.
- Critically analyze how more than one kind of ML algorithm (for instance, Random Forest, logistic, SVM) can perform in identifying ERP specific intrusions.
- Cloud ERP systems produce an enormous amount of data that is structured and unstructured in nature. What is more, ML models can successfully manage such complex data sets.
- There is always a potential intruder on any network or executing some malicious activity on any computer system, this is where Machine Learning models shine, for they are used in monitoring and extracting patterns and anomalies from the network or system activities in real-time.

2. Literature review

2.1 How is Cloud ERP Systems implemented with Machine Learning?

Investigating the use of an Artificial Intelligence Model in an ERP Cloud-Based System by (Nikitha Yathiraju) investigates the application of artificial intelligence to cloud-based ERP systems and understanding how such technologies can be applied to the analysis and forecast of systems data, as well as their automation in ERP systems. The study uses machine learning algorithms including decision tree, neural network and reinforcement learning models. It challenges their ability to bring expertise in addressing decision support, as well as data quality issues in a cloud-based ERP system. The assessment

is conducted using such factors as prediction accuracy, computational complexity, and changing input data. The AI models demonstrated high advantages in terms of speed and general data accuracy, in comparison to the conventional ERP systems with integrated AI. In the examined models, the neural networks delivered the highest accuracy, but the decision trees were again faster in computation. Thus, several limitations of the study include a high initial computational cost of developing AI models and the poor integration of such models into existing ERP systems. It also points out that the accuracy of the models depends on the quality of training data fed to AI models.

Ebrima Jaw and Xueming Wang researched on methods of ensemble learning in increasing the technique of intrusion detection for cloud-based ERP systems. In their study, they dedicated their study methods like bagging, boosting, and stacking to increase prediction and reduce the false positive value. The researchers used techniques such as AdaBoost, gradient boosting, and stacked generalization for utilizing a cloud ERP dataset. Self-generated results highlighted that the highest performance was attained through stacked generalization thereby being precise, computationally efficient accuracy standing at 97.5 %. However, AdaBoost obtained a slimmer accuracy margin, but at a much higher computational cost. The study also underscored that ensemble methods help to achieve a fairly optimal solution in terms of intensity of resource consumption and speed in environments in the cloud kind. However, it has some limitations including only focusing on a single ERP system where an investigation of other cloud platforms and different ERP systems would prove to be complex. Furthermore, it was also seen that the management of ensemble methods across several types of clouds is quite challenging.

Farnaz, N., & Jabbar, M focuses on assessing the ability of the proposed Random Forest (RF) model in detecting network intrusions in cloud environments. This research employs the NSL-KDD dataset to show how well the RF model performs based on factors like the accuracy of the result and false positive values as compared to the conventional models like the decision tree models. The experimental study also indicated that the RF model indeed has a higher detection efficiency while also possessing less false positives and as such is best suited for use in cloud-based ERP systems where the security issues function as paramount importance. Lack of new datasets means that results and findings may not be truly relevant in the present day because attackers have since moved to better and more complex tactics.

2.2 Techniques and Applications for Intrusion Detection System (IDS)

Muhammad Ashfaq Khan, Yang woo Kim study is about designing an enhanced IDS for protecting cloud computing domains by employing a combined deep learning technique. They employ IDS as the initial stage and desire to boost the detection precision and diminish false positives on IDS. Besides, the paper proposes a novel double-network system which integrates CNNs and RNNs jointly; CNNs is expected to proceed feature extraction while RNNs is for sequence learning. The data set used is network traffic data and the performance of the model is measured using detection accuracy and precisions, recall, F1-measure. The hybrid method is then compared to conventional deep learning structures such as standalone CNNs and RNNs in doing so, the authors showed that the hybrid deep learning model achieved much higher detection rates while at the same time, providing lower false positive rates compared to standalone CNN and RNN architectures. The proposed approach achieved a better F1 score and better generalization properties in a classification of diverse types of network attack and, therefore, can be considered a promising solution for real-time detection of intrusions in cloud IT infrastructures. Another limitation that the authors of the paper pointed out is the high computation cost needed to train the hybrid model. Also, the performance of the proposed model strongly depends on the quality and variability of the training data set what may hinder its practical usage in other environments.

Mahfuzul Riaz and Muhammad Arshad's work titled Intrusion Detection in Cloud Environment: A Comparative Study of the Approaches published in the International Journal of Advanced Computer Science and Applications (IJACSA) assess the existing network intrusion detection techniques for the

cloud computing systems. Its goal is to perform a comparative review of classic as well as new, modern approaches to intrusion detection, with reference to their viability in identifying threats pertinent to cloud-based systems. In this context, the study adopts systematic reviews to compare the various IDS approaches such as, signature-based IDS, anomaly-based IDS, and hybrid IDS. Besides people involved, performance measurement in detection is usually based on best-known benchmarks such as the detection rate, false positive rate, and necessary computational resources. The result of the study demonstrates that the proposed hybrid IDS models based on signature and anomaly-based models which are used as separate IDS's may be leading to a higher detection rate and low false positives in cloud environment. But authors indicate that such integration increases complexity and drives up the costs of construction, thus, makes them not very favorable in the context of cloud implementations with limited computational capabilities. The focus of the research is given to the development of lightweight adaptive IDS solutions that would be able to adapt to increased workload in cloud environments. However, fundamental to the study, the simulation data may not be able to capture the real-world cloud environment enough. The evaluation criteria are restricted to detection effectiveness and false positive rates without exploring more into factors like response time on the system or the overall effects on a network. However, the paper does not spend much time detailing the issues of applying IDS solutions in multi-cloud or hybrid cloud environments.

A study has been published by three authors Wenting Wang, Hongri Liu, and Bailing Wang in the journal Symmetry in the year 2023 with the goal of that research is to enhance the IDS abilities and accomplishments in cloud contexts using a combined machine learning model. To this end, the authors use hybrid methods which use support vector machines (SVM) and random forests. The researchers used the SVM model for the first level of anomaly detection, and the random forest classifier for second- level classification for improved precision. They performed the test using NSL-KDD dataset and compared their model with single SVM & random forest models. The performance measures used in this study were accuracy, false positive rate as well as the time taken to detect the anomaly. The proposed hybrid model was found to be superior to the individual classifiers in terms of having higher accuracy at 96.8%, and a low false positive value. The model also showed an improvement in the detection time, which makes it ideal where real-time detection of intrusions is required. The study also shows that it is possible to use multiple machine learning models for better detection in dynamic cloud environments. An obvious limitation is that of the use of a model that requires a pre-labelled dataset, and which may not therefore address real-life variations as effectively. However, the time required to compile and generate a hybrid instrumentation program from the Android code base as well as the overall computational overhead of the resulting hybrid instrument can be problematic in cloud scenarios with limited computational resources. The authors propose more testing involving real-time data to assess the effectiveness of the proposed model.

The work presented by Gao, X., Shan, C., Hu, C., Niu, Z., C Liu, Z in their paper aims an adaptive ensemble machine learning model for intrusion detection which adjusts the model's parameters with the fluctuating network traffic to improve the predictive capabilities of intrusion detection The ensemble model includes multiple classifiers; namely, Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN) using dynamic weighting. This plays a role in responding to dynamism in the cloud network environment. The adaptive model was found to outperform single classifiers when traffic was varying during the learning process, which is a characteristic of cloud ERP systems, thereby providing the desired level of adaptability. The integrated ensemble model leads to higher computational complexities which may become a disadvantage especially when it comes to real time detection in cloud environment with limited computational means.

2.3 The concept of cybersecurity in cloud environments and SaaS

Mamoona Humayun, Mahmood Niazi, Maram Fahhad Almufareh, N. Z. Jhanjhi, Sajjad Mahmood, and Mohammad Alshayeb have discussed security threats of Software-as-a service (SaaS) and give recommendations on how to prevent them in their journal applied sciences. The intention is to present a clear picture of security threat landscape in the SaaS model and the specific actionable strategies that can be implemented in viable SaaS deployments. In the present paper, the data are collected through a multivocal literature review (MLR) that employs both academic and grey literature, including peer-reviewed articles, industry reports, white papers, and best practice guidelines. Combination of researchers and practitioners from the field is beneficial due to the application of the MLR method which gives an opportunity to have a comprehensive view on SaaS security. The study classifies these security challenges in a systematic manner and gives risk avoidance advice on these risks. This paper reveals main potential security issues in SaaS including data leakage, weak identity administration, and inadequate encryption. It also discusses the absence of consistency in how organizations approach protection of their SaaS applications as a problem. To overcome the challenges, the study proposed the following use of best practices: The use of sound encryption techniques, the use of multi-factor authentication (MFA), and the adherence to the ISMS standards of security systems. Finally, the research focuses on the carried-out monitoring and auditing as the essential daily practice for preserving SaaS environment security. A drawback of the study is that the multivocal approach is comprised of sources other than scholarly articles and journals that might not be as strict. This could permanently settle certain bias in the choice of the data as well as in the process of their analysis. Further, the work highlights generalized recommendations for enhancing cybersecurity without thorough examination of effectiveness of these measures between distinct types of SaaS applications as well as between various industries.

Samuel Oladiipo Olabanji, Chinasa Susan Adigwe presented their research that proposes to analyze the possibility of using the users' behavior analysis incorporated into the AI-based cloud security systems. This is because it explains how behavior patterns can be used to enhance threat detection in the cloud, owing to new complex threats that are constantly being developed. In this case, the research incorporates the use of UBA together with Machine learning algorithms in identifying various dispositions of user activities. On an evaluation of several models within a simulation of the cloud environment, the models that have been considered are the decision tree, the support vector machine (SVM), and neural networks. Using data from actual users' activity logs the effectiveness of the models is evaluated in terms of separating begin behaviors from threats. The provided models are compared using performance measures including precision, recall and overall detection rate. This means that models with UBA are underlined to yield massive improvement in enhanced accurate outcome detection, especially in complex and concealed risks. Incorporation of behavior analysis in the neural network model of activity recognition showed the best performance in recognizing anomalous activities with the lowest false positive rate. In this study, the focus is on the results of investigations on the incorporation of user behaviors around these intelligent systems as the primary imperative for improving AI model accuracy when anticipating cyber threats in the cloud infrastructure. There is a limitation to the fact that the design of the study relies on historical user behaviors data that may not dynamically change or effectively respond to new or insignificant attack patterns. The study also highlights the problems of how to protect the user's anonymity while still being able to collect behavioral data. In particular, the complexity of introducing these artificial intelligence driven models could be a problem for cloud models that are smaller and have more limited resources. The study provides recommendations to engage the use of privacy preserving mechanisms for gathering UBA data.

2.4 A Comparative and Survey study on Machine Learning Techniques for IDS

A survey of data mining and machine learning methods for cybersecurity intrusion detection by Buczak, A. L., C Guven, E seeks to present a literature review on different data mining and machine learning methods used in cybersecurity in the context of IDS. Thus, the paper will cover different approaches to unsupervised learning such as decision tree, SVM and clustering for anomaly intrusion detection in the network traffic. The study also reveals that learning algorithms, random forests in particular, achieve better results compared to a single classifier used for distinguishing between diverse types of network intrusions. Some of the issues that can be identified from the review include how the algorithms perform in the situation whereby the datasets are imbalanced and how the algorithms change in response to several types of attacks. The study also discusses such prominent issues as the capability to perform the required calculations in real time, which is an issue in most of the models.

Comparison of deep learning method to traditional methods for network intrusion detection by Dong, B., & Wang, X aims at ascertaining the efficiency of deep learning models including CNNs with traditional methods including decision trees for uses in IDS. This results in the assessment of a few parameters and measures such as accuracy and precision of the models under analysis, taking place based on KDD CUP 99 dataset. There results revealed that technique is more effective in detecting multiple layers of attacks, where CNNs also outperformed other deep learning models in identifying multiple layers of attacks, thus indicating suitability in cloud ERP security. It can also be disadvantageous, because deep learning models require considerable amounts of computational resources and hence efficiency, typical for cloud settings, may be compromised.

A comparison study by Biswas, S. K establishes the capability of Naïve Bayes, SVM, and neural network models in the intrusion detection in cloud-based networks. The paper employs an NSL-KDD dataset then evaluates the performance of the models based on efficiency, accuracy, precision, recall and other related features. SVM and neural networks offered high detection rates than the other methods while Naïve Bayes used least time to complete its detection hence it is suitable for real time intrusion detection. Since the study relies on a single dataset, which reduces its relevance to modern, qualitatively different threat landscapes.

Adversarial machine learning for network intrusion detection systems survey by He, K., Kim, D. D., & Asghar, M. R examines the threats introduced by adversarial attack to machine learning based IDS with a focus on cloud environments. It reviews several adversarial machine learning strategies such as GANs and adversarial training to assess their performance in attacking and defending IDS models, which the work discovers particularly yields higher model robustness using adversarial training against complex attacks but at a cost of slower detection speed. The major disadvantage of employing adversarial defenses is that they add computational cost to IDS models that may not be desirable for real-time cloud ERP deployment.

Thus, the literature review focuses on the missing link between Machine Learning (ML) and the enhancement of Intrusion Detection Systems (IDS) for cloud-based ERP systems. Research is conducted covering various algorithms such as decision tree, support vector machine, neural network, boosting and stacking where each shows some differences in terms of accuracy, detection rate, and the time required for analysis. These models are useful because deep learning and hybrid models improve detection while reducing false positives, though the approaches can be computationally expensive. There is a focus on privacy-preserving methods and adaptability of security measures, solutions, and strategies to address new and changing threats and maintain the viable security of ERPs in everchanging cloud contexts.

3. Methodology

The process of creating an IDS using machine learning is systematic, which involves the use of various tools and technologies to mimic real-life issues. It includes data input and processing, model deployment and updating, and real time alerts and logging.

In this proposed work, since it was not possible to incorporate an actual ERP system due to copyright issues, Microsoft Forms was used to mimic an ERP tool. Microsoft Forms can also be used as a practical replacement for gathering input data as it mimics the role of an ERP in terms of offering traffic information such as duration, type of protocol, and type of service. The structure of the form is good because it has input validation and has many dropdown lists used to maintain uniform format and quality of the data. This simulation enables the system to show the possibility of implementing intrusion detection in an ERP like system.

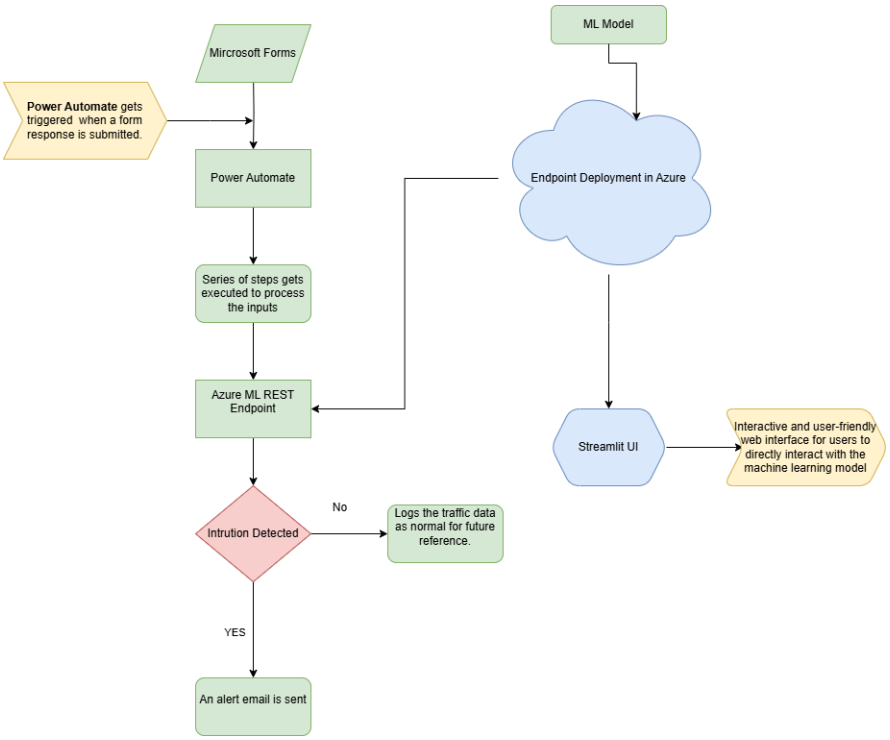

Figure 1: System architecture for proposed Intrusion Detection System.

So once the form submission is initiated, Power Automate then oversees the subsequent processing. It extracts the form responses, transforms it for analysis including encoding categorical variables for feature engineering and transforms the analysis output into a JSON payload for the machine learning model. Machine learning automation pipeline is achieved by passing pipeline the structured payload to the deployed ML model endpoint in Azure to give real-time prediction.

This machine learning model is a Random Forest classifier built on pre-processed network intrusion data in Azure ML Studio as a REST endpoint. This model is used to identify whether the network activity is an intruder, or whether the network traffic is normal. The next actions are deduced from the predictions made by the parsing of the data analysed by Power Automate. Of import is the alert email that is sent when an intrusion is detected on an account. Otherwise, the traffic data is recorded and stored for other analysis just like the regular traffic.

Furthermore, Streamlit UI is integrated into the project to demonstrate how clients can directly use the ML solution. Although Microsoft Forms emulates the GUI of the ERP tool, the Streamlit GUI is designed

to help users try out the model in a live environment. This also makes the system more easily demonstrable and, in general, easier for non-technical stakeholders to grasp as it takes away the complexities of the technical backend and distills them into easily consumable tables with simplified entry points for certain sections of the program. As a result, this methodology uses Microsoft Forms that, when linking with Power Automate and Azure ML, presents the concept of using machine learning to detect intrusions in contexts resembling ERP systems while functioning as a simulation tool. This is because this approach shows how the proposed solution can be implemented scalable and flexible to fit in real- life applications to identify anomalous situations.

4. Design and Implementation

The design and implementation of the Intrusion Detection System (IDS) involved a structured and iterative process. This allowed the integration of several tools and technologies into the mix to produce a highly competent system that would enable the identification of intrusion in a simulated environment of an ERP system. These activities entailed data simulation, automation of data, machine learning models, and designing easy to use interfaces.

4.1 Machine Learning Model Construction

The Machine learning model was built in Azure Notebook in Azure ML Studio using data for classification of traffic as ‘Normal’ or ‘Intrusion.’ The dataset was pre-processed to remove the missing values, convert factors such as protocol type and services to numeric values, and normalize other continuous variables like source bytes. In this phase, Exploratory Data Analysis was performed to determine attributes that are significant in the identification of intrusions and to balance the data for training purposes efficiently. The issue was posed as a binary classification problem where all the variations of the intrusions were grouped together into the ‘Intrusion’ class to avoid the occurrence of overly extensive false negatives.

Because of its high accuracy and interpretability, a Random Forest Classifier was employed, with hyperparameters being tuned by grid search. The trained model was re-deployed as a real time REST endpoint using Azure ML Studio to be integrated with automation process easily. This model of finding precision, accuracy and recall is used as the base for the Intrusion Detection System to accurately identify the hostile activities on a network.

4.2 System Design

The overall system also incorporated scalability to match a realistic implementation of an ERP system. This design sought to demonstrate the possibility of using machine learning models for intrusion detection regardless of the lack of commercial ERP system.

4.2.2 Data Simulation Using Microsoft Forms

As inaccessibility of an actual ERP tool was in question, Microsoft Forms was used as an ERP mimicking tool for obvious reasons. The form served as the primary interface for data input, mimicking the role of ERP systems by collecting critical attributes such as:

1. **Duration (ms):** Numeric field for connection duration of the call.
2. **Protocol Type:** A set of sub options next to the word ‘Protocol’ includes TCP, UDP and ICMP, among others.
3. **Service Type:** Menu drop down list with available services like HTTP, FTP and SMTP.
4. **Source and Destination Bytes:** Number fields holding measure of traffic across the network of the company.
5. **Wrong Fragment Count:** Numeric dropdown with an error handling bucket.

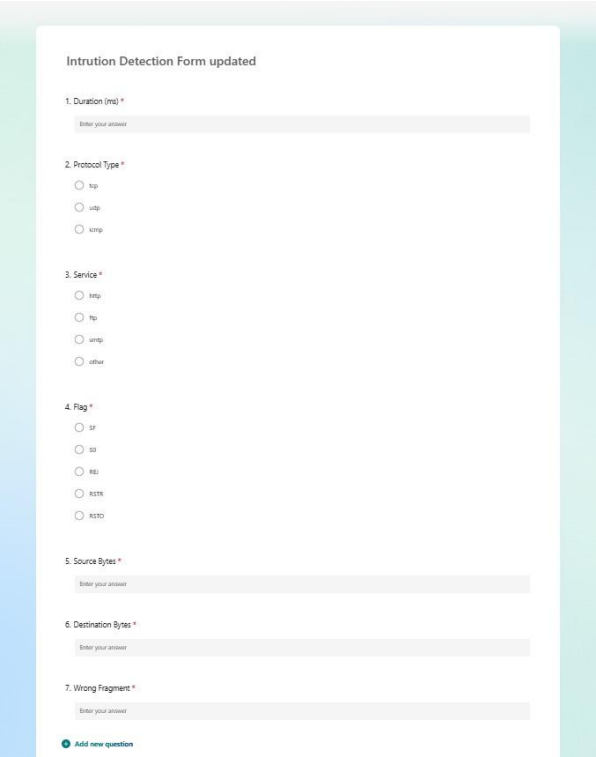

Figure 2: Data collection using Microsoft forms.

The form that was used made sure that high data quality was achieved by incorporating the use of input validation and the use of dropdown list. This approach showed that it is possible to mimic traffic data with real user input of actual ERP system while being a flexible solution in organizations that do not have access to ERP.

4.2.2 Data Flow Design

The data flow design incorporated a clear and logical sequence of actions:

1. **Input Collection:** The information filled in the Microsoft Forms.
2. **Processing:** On the same category of data entry, using Power Automate, it is possible to oversee the form responses on an automated basis.
3. **Prediction:** To the Azure ML REST endpoint, data formatted as JSON payload and passed on for analysis.

4. **Response Handling:** The processed results from predictions, including notification and alert generation for intrusions, or logging of normal traffic.
5. Modularity was also adopted in this design to make it an easily scalable and flexible design that can fit several real-life situations.

4.3 Implementation Steps

The process further resembled the designed layout by using Microsoft Forms, Power Automate, and Azure ML in the process.

4.3.1 Input Data Simulation

Microsoft Forms was used to design a format with which data was required to be entered. Key steps included:

- Designing fields for essential attributes, including dropdowns for categorical variables and numeric validations for continuous data.
- Setting required fields to avoid incomplete data submissions.
- Using dropdowns for attributes like protocol type and service to enforce standardized input.

4.3.2 Power Automate

Fully Automated Workflow Power Automate served as the automation pipeline to process and route data:

1. **Trigger Setup:** A flow was triggered whenever a new response was submitted in Microsoft Forms.
2. **Fetch Responses:** The "Get response details" action dynamically retrieved input data.
3. **Data Transformation:**
    - Variables are initialized for each field, such as protocol type, service, and flag.
    - Categorical data encoded using Switch actions, mapping user inputs (e.g., TCP → 0, UDP → 1).

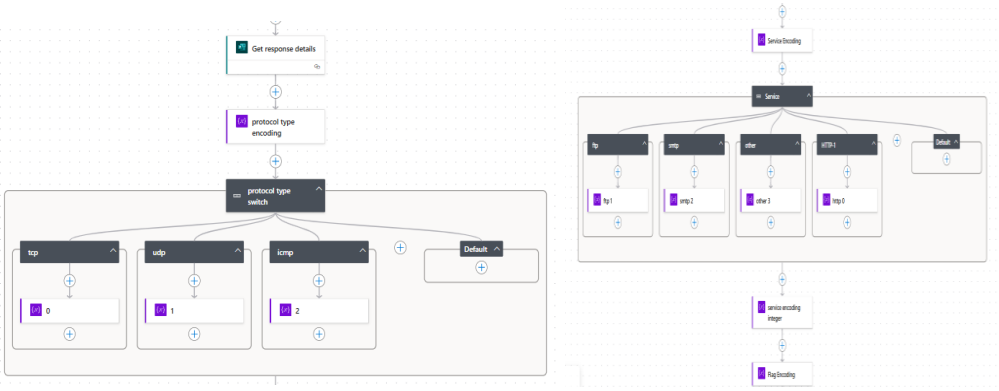

Figure 3: Workflow design in power automate.

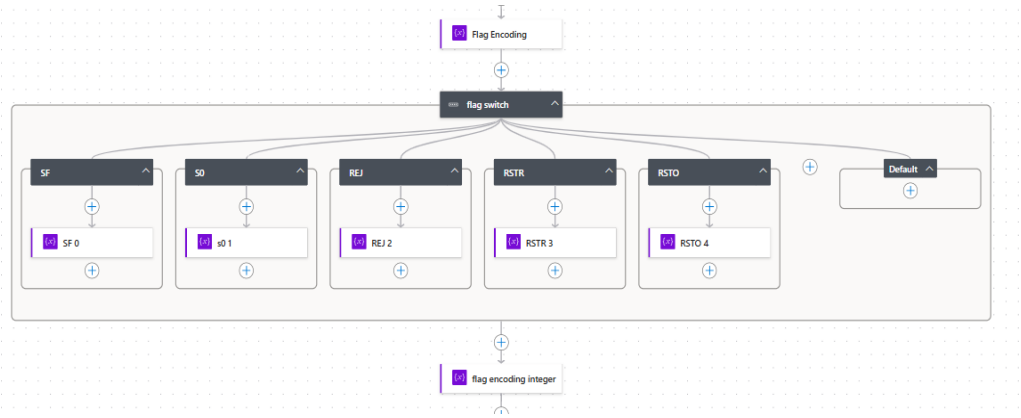

Figure 4: Workflow design in power automate.

4. **Data Formatting:** The Compose action aggregated all input and default data into a JSON payload.

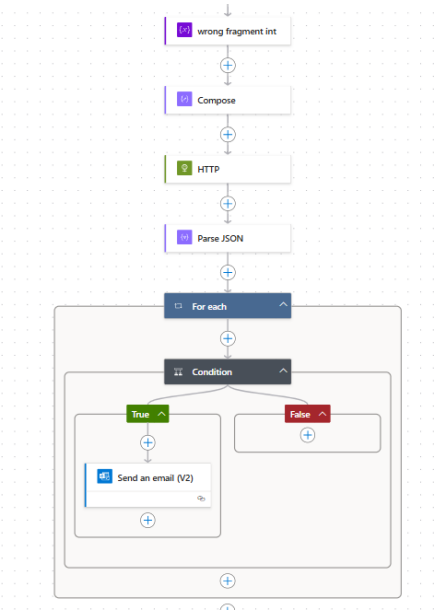

Figure 5: Workflow design in power automate.

4.3.3 Machine Learning Model Deployment in Azure

A pre-trained Random Forest classifier was used to classify network traffic as normal or intrusive:

- **Model Training:** Built on a cleaned intrusion dataset, the model was optimized for high accuracy.
- **Model Deployment:**
    - Uploaded the model to Azure ML Studio.

    - Configured deployment settings such as compute type (Container Instance) and generated a REST endpoint with a scoring URI and API key.

- The REST endpoint enabled seamless integration with Power Automate for real-time predictions.

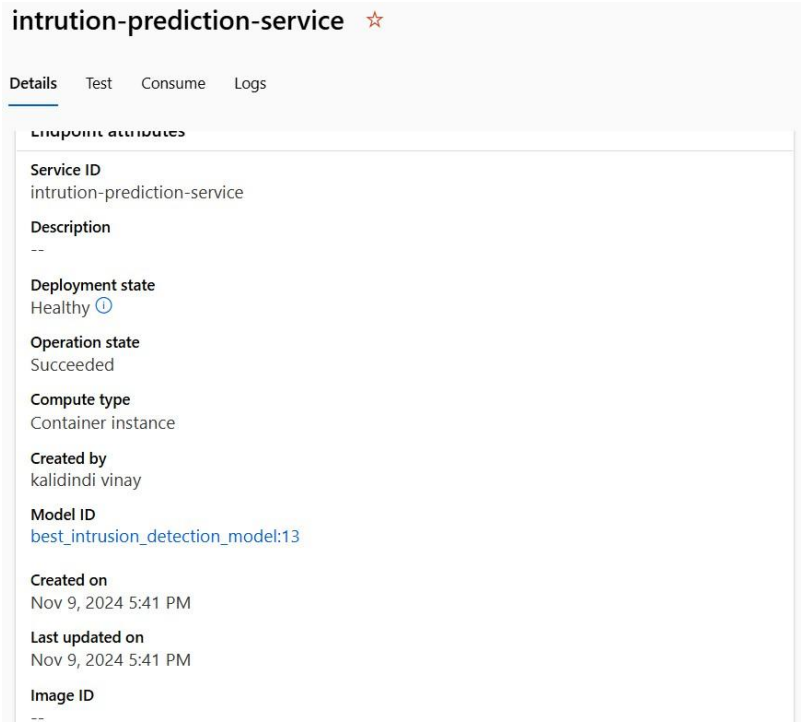

Figure 6: Azure endpoint deployment

4.3.4 Real Time User Interface with Streamlit

Streamlit provided an interactive interface for end users to directly interact with the ML model:

- Enabled direct entry of network traffic attributes thus avoiding the use of Microsoft Forms for convenience testing.
- Provided their prediction results in an understandable format for the user.
- Ensured that the new system was also within the comprehension of other users other than technicians by hiding backend details.

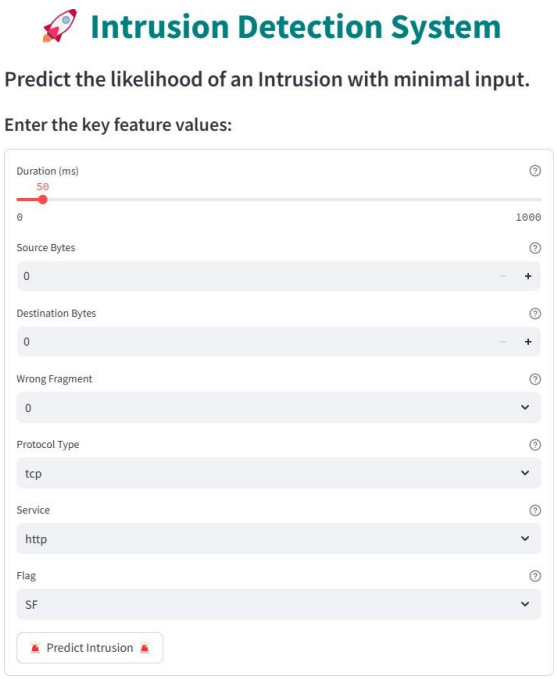

Figure 7: Input collection using Streamlit UI

4.4 Integration and Testing

The system underwent integration and testing to ensure smooth operation:

- **Data Input Validation:** Submitted test responses in Microsoft Forms to confirm proper flow initiation and data retrieval.
- **Pipeline Testing:** Validated each Power Automate step, including encoding, formatting, and HTTP requests.
- **Endpoint Validation:** Confirmed the Azure ML endpoint received inputs in the correct format and returned accurate predictions.
- **User Interface Testing:** Ensured the Streamlit UI functioned as expected, providing intuitive experience for stakeholders.

5. Results and Discussion

5.1 Machine Learning Model Performance

Machine learning model development was helpful as it produced valuable information. The Random Forest classifier was found to be the most accurate with accuracy of 99.94%, precision of 99.97%, and recall of 99.88% testifying to the model's efficiency in distinguishing normal and bad network traffic.

5.1.1 Dataset Distribution

As it can be seen in Fig. 1, the distribution of target classes in the training data indicates that the most numerous categories are "normal" with 60.33% of the data points, and the largest intrusion category is

“Neptune” with 35.594%. There were some classes with less than 10 samples such as ‘rootkit,’ multihop,’ and ‘spy.’ Still, the Random Forest model was able to accurately classify both the frequently and the less frequently occurring categories.

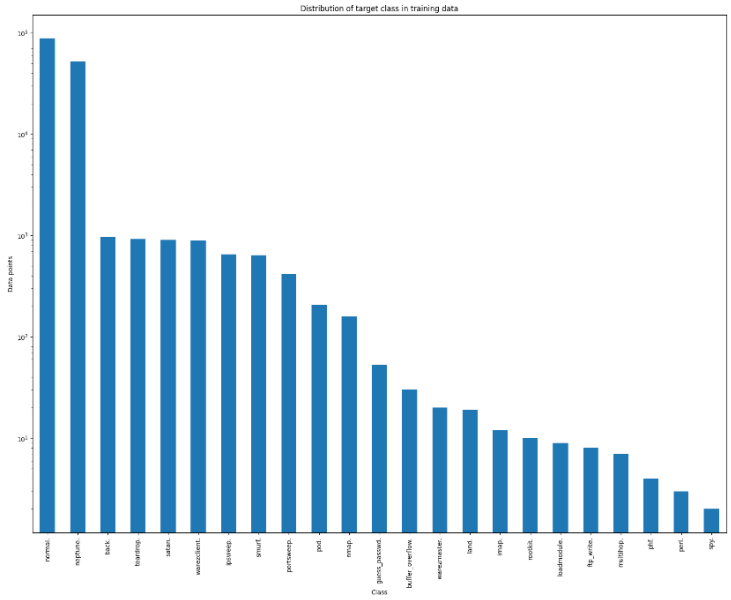

Figure 8: Distribution of target class in training data

5.1.2 Model Comparisons

The following confusion matrices for three models, Random Forest, Logistic Regression, and SVM, also support the Random Forest model. As seen from Figure 2 below, Random Forest had almost zero errors with only seventeen misclassifications while Logistic Regression and SVM faced difficulties especially in dealing with the rare classes.

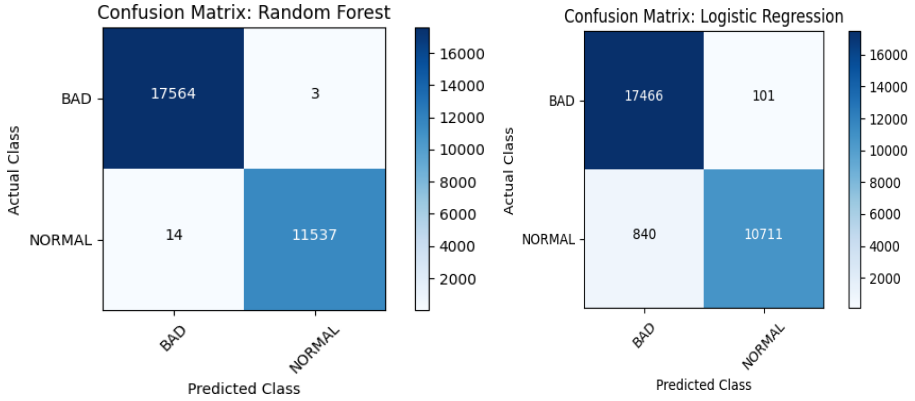

Figure 9: Confusion matrix for random forest & logistic regression.

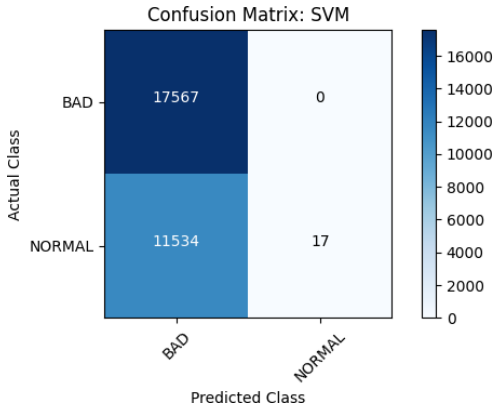

Figure 10: Confusion Matrix for SVM.

When it comes to model performance comparison, Random Forest appears to be the most accurate classification model with 99.94% accuracy, 99.97% precision and 99.88% recall thus being quite dependable for the intrusion detection task. Logistic Regression yielded satisfactory results though the routing could be categorized under this misclassification as it has a lower recall of 92.72% out of 96.76% accurately. Accuracy for SVM turned out to be extremely low at a level of 60.38%, although it demonstrated the maximum attainable precision regarding certain classes. As seen in Figure 3, these results prove that Random Forest is dependable for this chosen Intrusion Detection System.

```
Comparison of all models:
                Model  Accuracy  Precision    Recall
0       Random Forest  0.999416   0.999740  0.998788
1  Logistic Regression  0.967683   0.990659  0.927279
2                 SVM  0.603888   1.000000  0.001472
Best model (Random Forest) saved with accuracy: 0.9994161686929047
```

Figure 11: Model performance comparison.

5.1.3 Binary Classification Method

This project assumed intrusion detection as a binary classification problem. The first objective was to classify normal connections (positive class) from intrusions (negative class). This decision was because it is especially important not to classify intrusions as normal connections, which are a major security threat. Indeed, the Random Forest model, which is an ensemble model, helped to overcome the problem of an unbalanced dataset.

5.2.1 ERP simulation using Microsoft Forms

Owing to the current ERP tool accessibility being limited by the commercial aspect, Microsoft Forms was used to replicate an ERP setting. It also included important input parameters such as duration, protocol type, and service type. Dropdowns were used and input validations to guarantee stable and standard quality of the data entered for testing in real-time.

5.2.2 Data processing using Power Automate

Form responses were efficiently processed by Power Automate. Key tasks included:

- Interactive responses from the form.
- Categorizing variables for analysis and converting them into numerical codes.

- Format of the data into the JSON payload format that should be passed to the Azure ML endpoint.
- As observed from the flow diagram in (Figure 5), this payload was passed to the deployed model by Power Automate.

The following actions were executed based on the model's predictions:

Intrusion Detected: An alert email was sent.

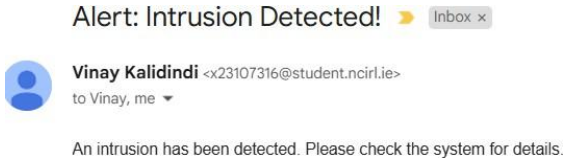

Figure 12: Real time mail notification.

No Intrusion Detected: The details of the traffic situation were recorded to be used later.

5.3 Real-Time Deployment and Streamlit Integration

To make the system easily testable in real time, a Streamlit web application has been created. This created ease of use interface that enabled direct interaction with the developed ML model. While working with Microsoft Forms, the authors were imitating the ERP interface, but with the help of Streamlit users were able to see how the ML model would work in practice in a test environment, connecting complex backend processes with a simple and straightforward frontend.

5.4 Discussion

The IDS developed in this paper illustrates that it is possible and practical to apply machine learning techniques to anomaly detection in ERP like scenarios. By incorporating Microsoft Forms, Power Automate, Azure ML, Streamlit the system was highly accurate and in real time.

5.4.1 Challenges and Future Directions

The approaches discussed can be challenged with certain questions: whether an enterprise should focus on a specific type of knowledge management; how to integrate the methods into an enterprise's structure: and what methods are essential or appropriate for its type of activity.

The project demonstrated how this kind of imbalance may be dealt with, but more sophisticated treatment could be necessary in conditions where class disparity is severe. This means that real-world deployment of such systems would require evaluating the techniques effectively against changing intrusion strategies and large diverse datasets.

6. Conclusion and Future Work

The proposed framework of an Intrusion Detection System (IDS) using machine learning has been successful in focusing on the paramount need for security on cloud ERP systems. The Random Forest classifier was used to identify normal and malicious traffic in which the system had a high accuracy of 99.94%. Further analysis included data preprocessing, feature engineering, and utilizing proper steps for instantaneous model evaluation. Microsoft Forms to create ERP-like simulation along with integration of Power Automate and Azure ML has given an effective, easy to use and reusable solution with real life applicability.

This project also addresses the problem of data imbalance in the intrusion set, the important thing being the minimization of false negatives that threaten organizations' security. Applying Power Automate for

the processes automation, Streamlit for the live testing improved overall flexibility and a wide approachable for non-Technological workforce.

However, there are some future work opportunities worth mentioning. Enhancements could include:

- Real-Time Data Integration: Integrating the designed system with actual real time ERP systems for intrusion detection.
- Advanced Anomaly Detection: The integration of the unsupervised learning models to classify unknown attack patterns in the network.
- Cloud Scalability: Specifying to more enhanced infrastructure such as Kubernetes or AKS for managing increased data traffic.
- Explainable AI: Introducing interpretability modules to enhance the explanation of model predictions and enhance their transparency.
- Performance Optimization: Tuning more of other existing techniques like Gradient Boosting or improving the deep learning models for better rates of recall and detection time.

References

N. Yathiraju, "Investigating the use of an Artificial Intelligence Model in an ERP Cloud-Based System," International Journal of Electrical, Electronics and Computers, vol. 7, no. 2, pp. 01-26, 2022.

Muhammad Ashfaq Khan and Yangwoo Kim, "Toward Efficient Intrusion Detection System Using Hybrid Deep Learning Approach," Symmetry, vol. 14, no. 9, p. 1916, Jan. 2021.

Wenting Wang, Hongri Liu, and Bailing Wang "Intrusion Detection in cloud computing by the approach of hybrid machine learning" Symmetry, vol 14 Feb 2023.

Mahfuzul Riaz and Muhammad Arshad "Intrusion detection in cloud environment: A comparative study of the approaches" International journal of advanced computer science and applications (IJACSA), vol. 15, no. 3, 2024.

M. Humayun, M. Niazi, M. F. Almufareh, N. Z. Jhanjhi, S. Mahmood, and M. Alshayeb, "Software-as-a-Service Security Challenges and Best Practices: A Multivocal Literature Review," Applied Sciences, vol. 12, no. 8, p. 3953, Apr. 2022.

Samuel Oladiipo Olabanji, Yewande Alice Marquis, Chinasa Susan Adigwe, Samson Abidemi Ajayi, Tunbosun Oyewale Oladoyinbo, and Oluwaseun Oladeji Olaniyi "AI-Driven cloud security: Examining the impact of user behavior analysis on threat detection" Asian journal of research in computer science, vol.17, issue 3, Jan 2024

A. Vinolia, N. Kanya, V.N Rajavarman "Machine learning and deep learning-based intrusion detection in cloud environment."

Wuqi Qi, Wei Wu, Hoa Wang, Lu Ou, Ning Hu, Zhihong Tian "Intrusion detection technique analysis in cloud computing" IEEE 12th International conference on cloud networking (Cloudnet)

Muhammad Salman Saeed, Raman Saurabh, Sarang Babasaheb Bhasme, Alexey N. Nazrov "Machine Learning Based Intrusion Detection System in cloud Environment."

Hanaa Attou, Azidine Guezzaz, Said Benkirane, Mourade Azrour, Yousef Farhaoui, “Cloud based intrusion detection approach using machine learning techniques” IEEE explore,2023.

Amna riaz, Hafiz Farooq, Usman Younis “Intrusion Detection system in cloud computing: A Contemporary review of Techniques and solutions,” Journal of Information science and engineering. Jan. 2017

Manisha Bharati, Sharvaree Tamane “Intrusion Detection systems & future challenges in cloud in cloud-based environment, 2017 1st International conference on intelligent systems and information management”

Bharathi Reddy S, Malathi D., and Shijoe Jose “An intrusion detection and prevention system in cloud computing: A Technical review” ARPN journal of engineering and applied sciences, Vol. 12, June 2017.

Q. Ma, C. Sun, and B. Cui, “A Novel Model for Anomaly Detection in Network Traffic Based on Support Vector Machine and Clustering,” Security and Communication Networks, vol. 2021, p. e2170788, Nov. 2021

S. Ahmadi, “Network Intrusion Detection in Cloud Environments: A Comparative Analysis of Approaches,” International journal of advanced computer science and applications (IJACSA), vol. 15, no. 3, 2024.

Muhammad Sajid, Kaleem Razzaq Malik, Ahmad Almogren, “Enhancing Intrusion detection: a hybrid machine and deep learning approach” journal of cloud computing July 2024

Selman Hizal, Unal Cavusoglu, Devrim Akgun, 2021 third international congress on Human-computer interaction, optimization, and robotic applications (HORA)

Utsav Upadhyay, Alok Kumar, Satyabarata Roy, Umashankar Rawat, Sandeep Chaurasia “Defending the cloud: Understanding the role of explainable ai in intrusion detection systems” 2023 16th international conference on security of information and Networks (SIN), Nov 2023.

Manu Kohli “Using machine learning algorithms on data residing in sap ERP application to predict equipment failures” International journal of engineering & technology.