Mining Network Intrusion Patterns

Jagrat Patkar
Masters of Science in Data Science
University of Colorado Boulder
jagrat.patkar@colorado.edu

ABSTRACT

The increasing frequency of cyber-attacks necessitates the development of effective intrusion detection systems to safeguard network infrastructures. The KDD Cup 1999 dataset, representing diverse network interactions, offers a practical resource for the development and assessment of intrusion detection models. This project undertaken as a part of the ***DTSA 5506 Data Mining Project*** coursework, proposed to employ data mining techniques on the KDD Cup 1999 dataset to discover patterns and anomalies associated with network intrusions. The initial phase of the project was dedicated to meticulous data preprocessing, ensuring the dataset's consistency and normalization for precise analysis. The subsequent stages involved the application and evaluation of various data mining algorithms to understand the dataset's underlying structure and identify anomalies indicative of potential cyber threats.

Keywords

Data mining; Intrusion Detection.

1.INTRODUCTION

The domain of cybersecurity is persistently challenged by the advent of sophisticated cyber-attacks that threaten network infrastructures. A key aspect of cybersecurity is intrusion detection, which involves identifying malicious activities in a network to ensure timely countermeasures. This project proposal focuses on the analysis of the KDD Cup 1999 dataset through data mining techniques to discover patterns and anomalies associated with network intrusions, thereby contributing to the enhancement of intrusion detection systems.

The importance of this endeavour stems from the constant need to improve network security measures as cyber threats continue to evolve. By analysing network interactions and identifying traits indicative of intrusions, it becomes plausible to develop more effective intrusion detection mechanisms. This not only aids in promptly identifying potential threats but also in understanding the manner of cyber adversaries, which is significant for deriving robust security protocols.

Existing solutions in intrusion detection have made significant strides; however, they often suffer from limitations such as high false positive rates, inability to detect new or sophisticated attack vectors, and resource-intensive operations. Moreover, many of these solutions are built on predefined rules and heuristics which may not adapt well to evolving threats. The proposed project aims to address some of these limitations by employing data mining algorithms to the KDD Cup 1999 dataset, this dataset represents a wide range of network interactions and has been a influential dataset for intrusion detection research, featuring various types of attack patterns as summarised in [3]. These include Denial of Service (DoS), Remote to Local (R2L) attacks, User to Root (U2R) attacks, probing attacks for reconnaissance and detection, and data transmission data attacks. The dataset provides 41 characteristics, divided into three different categories: traffic characteristics, content characteristics, and other characteristics.

In light of the project's limited scope, the objective is to take a measured step towards understanding the potential of data mining in improving intrusion detection systems (IDS). Through this exploration, it is expected that insights will be gleaned on how data-driven approaches can be employed to identify network intrusions more accurately, thereby contributing to the broader goal of enhancing network security.

2.RELATED WORK

The integration of data mining techniques into the field of intrusion detection represents a significant advancement, offering sophisticated methodologies for identifying and analyzing security threats in network infrastructures. This section provides an overview of the diverse data mining techniques employed in intrusion detection systems (IDS), as derived from insights in relevant literature.

As delineated in the survey paper [2], data mining techniques applicable in the domain of intrusion detection are classified into seven distinct categories, each fulfilling a specific role in the detection and response to network threats:

- **Reinforcement Learning**: Essential for determining actions based on historical data, this technique enables IDS to adapt to new threats effectively.
- **Regression**: This method plays a crucial role in forecasting potential security incidents by predicting numeric or continuous values based on existing patterns.
- **Classification**: As a fundamental technique in IDS, classification assists in predicting the category (normal or malicious) of network behavior from a dataset.
- **Optimization**: Involving the search for optimal or satisfactory solutions through iterative execution, optimization is key to decision-making in IDS.
- **Ensemble Methods**: By combining multiple classifiers' predictions, these methods improve the accuracy and reliability of intrusion detection.
- **Rule System**: This approach uses a set of if-then rules to classify and identify network intrusions based on predefined criteria.
- **Clustering**: Clustering aids in identifying unusual patterns by grouping data into meaningful sub-classes, which can indicate security breaches.

Another survey paper [1] emphasizes the significance of data mining techniques in contemporary intrusion detection systems. It highlights the frequent application of Classification, Clustering, and Association Rules to obtain understanding about intrusions from network data. These techniques are crucial for highlighting underlying patterns and anomalies in network traffic, characteristic of potential intrusions.

The following subsections of this report will look into specifics of some of these algorithms, offering a detailed exploration of their application in intrusion detection. This analysis aims to provide a more comprehensive understanding of how mining techniques contribute to the detection and analysis of network intrusions, particularly in the context of the KDD Cup 1999 dataset. This examination will also help in identifying the most effective techniques for the project's objectives.

2.1.Classification

Classification in data mining is a very important technique in intrusion detection systems (IDS), enabling the categorization of network traffic into normal or malicious classes. According to the literature survey in paper [2], several classification algorithms have been thoroughly studied, each with its unique strengths and weaknesses. Support Vector Machine (SVM) is noted for being the most used and overall best classifier in the literature, known for its flexibility in parameter selection and ability to process high-dimensional feature vectors. However, it is also sensitive to noise and affected by the size and dimensionality of the dataset. Neural Networks, recognized for superior pattern recognition, perform specially well with prior knowledge and self-organizing maps. Their main drawbacks include prolonged training time and reduced precision in detecting less frequent attacks. Decision Trees, particularly the C5.0 algorithm, offer robust performance with missing data and large input fields, and are valued for their accuracy and reduced training time. Bayesian Networks are proficient when training data is scarce, offering high accuracy and simpler models that reduce time complexity. On the other hand, k-Nearest Neighbors (k-NN), though easy to implement, suffer from issues like high dimensionality and overfitting.

Paper [1] discusses classification in the context of IDS, highlighting that this approach involves a learning step to form a classifier, followed by the classification step where the model is used to predict class labels. The authors mention a paper which states that classification models are formed based on pre-labeled data, focusing not on discovering new classes but on categorizing new records into these predetermined classes. This technique is particularly effective in misuse detection, although its efficiency can be limited by the large volume of data required for analysis. The paper further discusses on a data classification process for intrusion detection, which involves training with pre-labeled sequences of system calls, scanning intrusion traces, and labeling sequences as normal or abnormal based on the presence in the normal list. This process underscores the formal approach of classification in distinguishing between normal and malicious network activities.

2.2.Clustering

Clustering, as a data mining technique, plays a significant role in intrusion detection systems (IDS) by categorizing data into groups based on similarity, thereby enabling the identification of unusual patterns that may signify network intrusions. Paper [2] provides an analysis of different clustering algorithms and their efficacy in IDS. K-Means, known for its simplicity and low computational complexity, is highlighted as particularly suitable for real-time IDS. However, its application is limited by the prerequisite of specifying the number of clusters beforehand and the need to input the value of *k* in advance. In contrast, Hierarchical Agglomerative Clustering is advantageous as it does not require prior knowledge of the number of clusters, making it more flexible in varying intrusion detection scenarios.

Paper [1] elaborates on the utility of clustering in handling large volumes of network data, which can be inconvenient for human labeling due to its time-consuming and costly nature. It mentions a reference stating that clustering algorithms can be categorized into four groups: partitioning, hierarchical, density-based, and grid-based algorithms. The paper highlights that clustering techniques are essential in uncovering complex intrusions over different time periods and are effective in both anomaly detection and misuse detection. Clustering, as an unsupervised learning mechanism, identifies patterns in unlabeled data with numerous dimensions, where the deviation of patterns from established clusters may indicate new or unusual activities, potentially flagging them as part of a unconventional attack. Furthermore, the authors mention a paper which introduced a basic methodology for identifying intrusions using clustering. This involves identifying the largest cluster and labeling it as normal, followed by arranging the remaining clusters based on their proximity to the largest one. Clusters comprising a significant portion of the data instances are also labeled as normal, while the others are marked as malicious. Post-clustering, heuristics are employed to label each cluster automatically as either normal or malicious. These self-labeled clusters are then utilized to detect attacks in separate test datasets, underscoring the practical application of clustering in adeptly isolating normal and malicious network activities in IDS. This process demonstrates the capability of clustering algorithms to streamline the categorization of network data, thereby enhancing the effectiveness of intrusion detection systems.

2.3.Association Rule Mining

Association rule mining is another technique in data mining for intrusion detection systems (IDS), focusing on uncovering correlations between various data elements within a network. The technique treats each attribute/value pair in network data as an item, with collections of these items forming item sets. The primary goal is to identify item sets that frequently appear within the network data, thereby revealing multi-feature correlations. Association rules, often formulated as "if-then" statements, are derived from these correlations in the dataset. Paper [1] mentioned a reference which highlights that the Apriori algorithm was the first scalable algorithm developed for this purpose.

The utilization of association rule mining in intrusion detection is multifaceted. According to the authors referenced in [1], the basic steps for integrating association rule mining into IDS include organizing network data into a database table, where each row represents an audit record and each column a field within these records. Intrusions and user activities often display frequent correlations, which are effectively captured by association rules. These rules are not static; they can evolve by continuously integrating new rules from subsequent analyses into the cumulative rule set from all previous runs. This dynamic nature of association rule mining allows for the modification to new patterns and behaviors in network data, enhancing the detection accuracy of IDS.

3.COMPLETED WORK

In context of the literature gleaned from (Section 4), the proposed project aims to apply and extend upon the data mining techniques previously used in intrusion detection research. Each proposed model will be contextualized within the framework of existing literature, aiming that the project not only replicates but also tries to contribute to the ongoing dialogue in this domain.

3.1.EDA and Data Visaulization

The project's exploratory phase, involving Exploratory Data Analysis (EDA) and Data Visualization, has helped in understanding the details of the KDD Cup 1999 dataset. Leveraging Python libraries such as Pandas, Matplotlib, and Seaborn, this stage successfully reflected the dataset's underlying structures and patterns. It covered detailed statistical analyses, preprocessing tasks like encoding categorical variables, and informative data visualizations.

3.1.1.Data Understanding

The project's initial focus was on understanding the KDD Cup 1999 dataset, a cornerstone for network intrusion detection research. This phase involved an in-depth analysis of various network traffic features, such as connection duration, protocol types, and byte transfers. A key finding was the significant class imbalances, it was observed certain attack types or normal behaviors were more frequent than others.

3.1.2.Data Preprocessing

The preprocessing stage was important for transforming the dataset into a format suitable for machine learning algorithms. This involved encoding categorical variables using label encoding. Key features such as *protocol_type*, *service*, and *flag* were converted into numerical formats.

3.1.3.Data Visualization

The project employed a range of visualizations to glean deeper insights into the dataset:

1. **Protocol Type Distribution in Attack Types**: This bar chart illustrated the frequency of different protocol types (TCP, UDP, ICMP, etc.) within attack records, highlighting exploited protocols in network intrusions.

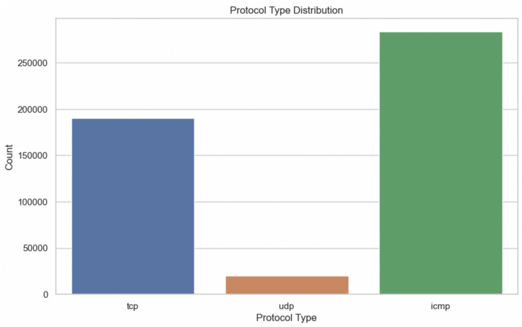

Figure 1. Protocol Type Distribution

2. **Top 10 Services Distribution in Attack Types**: A visualization focusing on the most common network services involved in attacks.

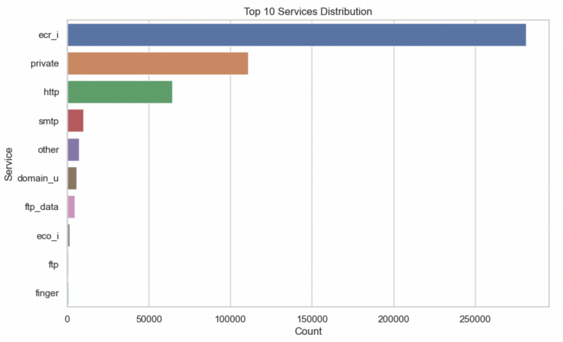

Figure 2. Top 10 Services Distribution

3. **Distribution of Attack Types**: A bar chart presenting the prevalence of various attack types (DoS, Probe, U2R, R2L), which helps understand the frequency of network intrusions.

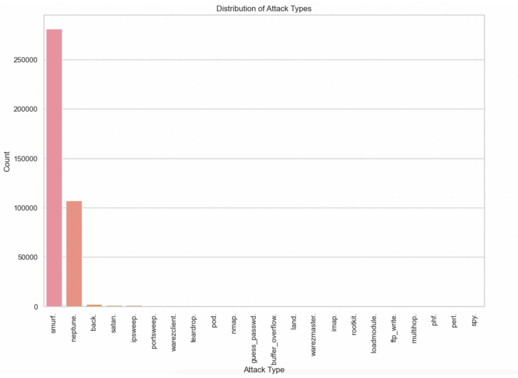

Figure 3. Distribution of Attack Types

4. **Distribution of Source Bytes (Log-Scaled, Positive Values Only)**: A histogram displaying the log-scaled distribution of *src_bytes*, revealing patterns in data transmission behavior.

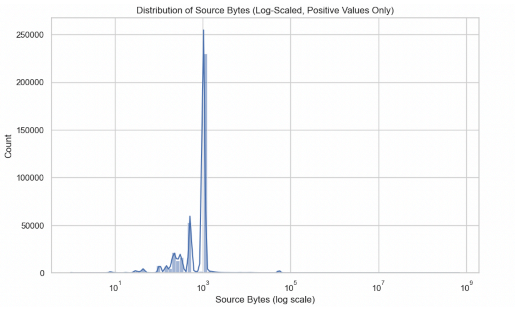

Figure 4. Distribution of Source Bytes (Log-scaled, Positive Values)

5. **Top 10 Services Distribution**: Similar to the (Figure 2), but applied to the entire dataset, this bar chart compared the frequency of top services across all network interactions.

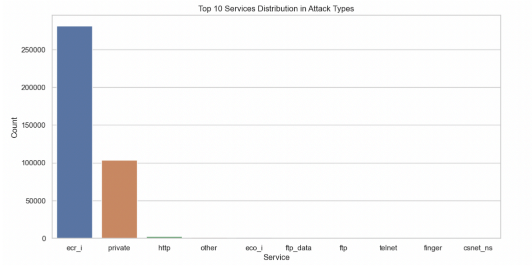

Figure 5. Top 10 Services Distribution in Attack Types

6. **Protocol Type Distribution**: This visualization showed the overall usage pattern of different protocol types in the dataset, irrespective of attack classifications.

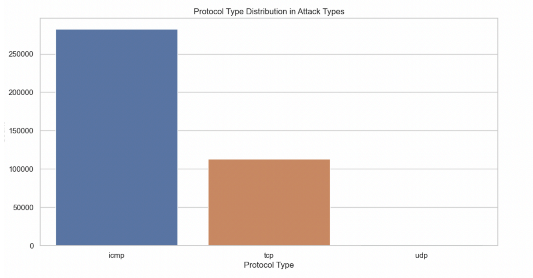

Figure 6. Protocol Type Distribution

3.2.Support Vector Machines (SVM)

In alignment with the literature highlighting SVM's proficiency in high-dimensional settings, the project applied this model to the KDD dataset's 41 features using Scikit-learn. The project conducted extensive hyperparameter tuning, particularly focusing on the regularization parameter *C*, the kernel coefficient *gamma*, and the *kernel type*. The best model was configured with *C* set to 10, *gamma* as '*scale*', and using the '*rbf*' *kernel*. This approach, yielded a model with high precision and robustness. The performance of the model will be discussed in (Section 4.1).

3.3.Random Forest Classifier

Aligning with previous research underscoring the efficacy of ensemble methods in intrusion detection, the Random Forest Classifier was selected for its resistance to overfitting and it's management of high-dimensional data. In the hyperparameter tuning phase, the model's performance was optimized by adjusting key parameters, specifically focusing on the number of trees (*n_estimators*) and the depth of each tree (*max_depth*). The optimal configuration, determined through rigorous testing, was found with *n_estimators* set to 200 and *max_depth* at 30. This configuration shows a preference for a more complex model, using a considerable number of trees and allowing substantial depth in each, to effectively capture the nuances and complexities of the dataset.

3.4.K-Nearest Neighbors (K-NN)

The application of K-Nearest Neighbors (K-NN) in this project was influenced by its noted flexibility when encountering new data, as highlighted in existing literature. The project's implementation involved fine-tuning key parameters, particularly the number of neighbors (*n_neighbors*) and the distance metric used. The optimal parameters identified were *n_neighbors* set to 3 and the distance metric chosen as '*manhattan*', reflecting a focused approach in considering the closest neighbors with the Manhattan distance metric for precise calculations. This tuning process addressing the high-dimensionality challenge inherent in the dataset, effectively leveraged K-NN's strengths.

3.5.Decision Trees

The implementation of the Decision Tree model in this project was guided by its value for interpretability in intrusion detection, as established in relevant literature. The model was tuned focusing on optimizing key parameters such as tree depth, *min_samples_split*, and *min_samples_leaf.* The best model configuration achieved was with a maximum depth of 30, a minimum of 1 sample per leaf, and a minimum of 2 samples required to split a node. This optimal configuration underlines the model's balance between complexity and interpretability, leveraging the Decision Tree's strength in generating clear and comprehensible decision rules from the dataset's intricacies.

4.EVALUATION

In the proposed project, the effectiveness of each model will be rigorously assessed using a set of well-established metrics, in line with methods employed in the literature. The primary metrics for evaluation will include accuracy, error rate, precision, and F1 score, each offering a unique perspective on the model's performance.

Accuracy, a commonly used metric in previous literature, measures the proportion of true results among the total number of cases examined. It provides a general idea of the model's overall effectiveness. On the other hand, the error rate, which is simply one minus the accuracy, reflects the proportion of all incorrect predictions made by the model. While both accuracy and error rate offer a broad view of performance, they may not always be enough, especially in imbalanced datasets where the cost of false positives and false negatives varies. To address this, precision will be used, which measures the proportion of true positives among all positive predictions. This metric is particularly crucial in intrusion detection, where the cost of false alarms (false positives) can be high. Lastly, the F1 score, which is the harmonic mean of precision and recall, will be used to provide a balanced measure of a model's precision and recall capabilities. The F1 score is particularly useful in scenarios where an balance between false positives and false negatives is essential. Together, these metrics will offer a comprehensive evaluation of each model's capacity to accurately and effectively identify network intrusions.

4.1.Support Vector Machine Evaluation

The evaluation of the SVM model in the project revealed a high performance across various metrics: the model achieved an accuracy of approximately 99.93%, precision of 99.93%, recall of 99.93%, and an F1 score of 99.93%. These metrics suggest a great level of accuracy in classifying network traffic into normal and various types of attack classes. The high precision indicates the model's success in correctly identifying positive cases, while the equally high recall shows its capability in capturing most of the actual positive cases. The F1 score, being the harmonic mean of precision and recall, supports the model's balanced strength in precision and recall.

4.2.Random Forest Evaluation

The evaluation of the Random Forest model showed encouraging results, underscoring its effectiveness in the context of network intrusion detection. The model achieved a remarkable accuracy of 99.97%, with precision, recall, and F1 score all closely aligned in the same range. These metrics suggest a high degree of reliability in the model's predictions.

4.3.K-Nearest Neighbors (KNN) Evaluation

The evaluation of the K-Nearest Neighbors (KNN) model demonstrated notable effectiveness in intrusion detection, reflected in its high accuracy of 99.95% and closely aligned precision, recall, and F1 score. These metrics indicate a strong ability of the KNN model to correctly classify both normal and attack instances.

4.4.Decision Tree Evaluation

The Decision Tree model displayed robust performance in the intrusion detection task, achieving an accuracy of around 99.95%, with similarly high precision, recall, and F1 score. These results suggest a strong capability of the model to distinguish between normal and attack instances effectively. The optimal parameters of *max_depth: 3*0, *min_samples_leaf: 1*, and *min_samples_split: 2*, indicate a preference for a deep tree structure, enhancing the model's ability to capture complex patterns in the data.

4.5.Interpretation of Results

Interpreting the results of the various models applied to the intrusion detection dataset requires a nuanced approach, considering both their performance metrics and potential limitations. While models like SVM, Random Forest, KNN, and Decision Tree demonstrated high accuracy, precision, recall, and F1 scores, caution is necessary before concluding their effectiveness in real-world applications. Key considerations include:

1. **Overfitting**: High performance could be indicative of overfitting, particularly in models with complex structures like Random Forest and Decision Tree. This would mean they are overly tuned to the training data and may not perform as well on unseen data.

2. **Class Imbalance**: The dataset's class imbalance might lead to skewed results, with models potentially performing well on the majority class but poorly on underrepresented classes. This imbalance could inflate overall performance metrics, giving a false impression of effectiveness.

4.6.Algorithm Selection for IDS

Intrusion Detection Systems (IDS) require efficient and effective classification algorithms to promptly detect and respond to potential threats. The choice of the right algorithm involves balancing various factors training time is a critical element due to the dynamic nature of network environments and the need for rapid adaptation to emerging threats. This section discusses the training times of the best models for each of the four algorithms and which model is most suited for IDS based on it:

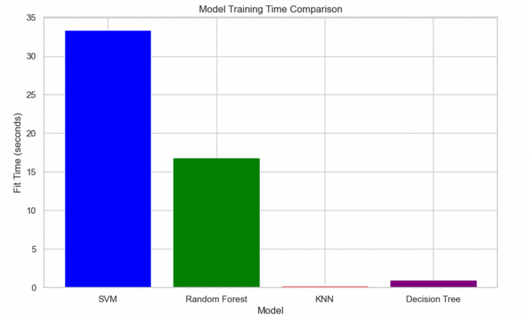

Figure 7. Model Training Time Comparison

1. **Support Vector Machine (SVM)** best model took approximately 33.36 seconds to train. SVM's relatively longer training time could be a concern in scenarios requiring frequent retraining or updates, such as adapting to new types of network attacks.
2. **Random Forest** best model training time was around 16.82 seconds. Despite its robustness, Random Forest's training time is moderate. In IDS, where models might need to be updated regularly, this could be a potential limitation.
3. **K-Nearest Neighbors (KNN)** best model training was notably fast at approximately 0.27 seconds. KNN stands out for its exceptionally fast training time, making it highly suitable for IDS that require quick adaptation to changing network patterns.
4. **Decision Tree** best model training time was approximately 1.00 second. Decision Tree offers a good balance between training speed and model complexity, which can be advantageous in dynamic IDS environments.

KNN, with its extremely fast training time, emerges as a the appropriate model (out of the 4) for IDS where training time is critical.

5.DISCUSSION, FUTURE WORK AND KEY FINDINGS

This three-week project successfully applied machine learning techniques to the KDD Cup 1999 dataset for intrusion detection, focusing on models such as SVM, Random Forest, K-NN, and Decision Trees. The initial week was dedicated to exploratory data analysis and data visualization, a crucial step for understanding and preparing the complex, high-dimensional dataset. The subsequent weeks involved intensive modeling, tuning, and evaluating these models. Key findings revealed high performance across models, with K-NN standing out for its rapid training time. However, challenges such as class imbalance within the dataset and the trade-off between model complexity and performance were notable. These aspects potentially influenced the reliability of performance metrics and the models' ability to generalize.

Looking ahead, future work in this area could focus on addressing the class imbalance through methods like synthetic data generation, applying more rigorous evaluation metrics like stratified cross-validation, and exploring additional models, including deep learning approaches. Testing the models with real-world data and further research into feature engineering could provide insights into improving model robustness and computational efficiency. Such advancements would be important in enhancing the effectiveness and applicability of intrusion detection systems, ensuring they remain adaptable and reliable in the face of evolving network threats.

6.CONCLUSION

The project report successfully outlines a comprehensive approach to enhancing intrusion detection using data mining techniques on the KDD Cup 1999 dataset. The initial phase, now completed, involved thorough Exploratory Data Analysis (EDA) and data visualization, providing deep insights into the dataset's intricacies. The next task consisted successful training and evaluation of several models: Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (K-NN), and Decision Trees. Each model displayed notable performance, with evaluation metrics such as accuracy, precision, recall, and F1 score highlighting their effectiveness in classifying network intrusion types. The SVM, in particular, showed exceptional performance, while K-NN was distinguished for its rapid training time, an essential factor in intrusion detection systems.

Completed within a 2-3 week timeline, this project not only replicated existing research but also contributed new analysis and benchmarks of models on the KDD Cup 1999 dataset. Future work can extend these findings by exploring methods to further reduce class imbalance, employing more rigorous evaluation metrics, and considering additional models, including deep learning approaches. This project underscores the dynamic nature of cybersecurity threats and the continuous need for adaptable, efficient, and robust intrusion detection systems in this ever-evolving landscape.

7.REFERENCES

1. Denatious, D.K. and John, A. 2012. Survey on data mining techniques to enhance intrusion detection. 2012 *International Conference on Computer Communication and Informatics*. http://dx.doi.org/10.1109/iccci.2012.6158822.
2. Salo, F., Injadat, M., Nassif, A.B., Shami, A., and Essex, A. 2018. Data Mining Techniques in Intrusion Detection Systems: A Systematic Literature Review. IEEE Access 6, 56046–56058. http://dx.doi.org/10.1109/access.2018.2872784.
3. Zhu, Y., Gaba, G.S., Almansour, F.M., Alroobaea, R., and Masud, M. 2021. Application of data mining technology in detecting network intrusion and security maintenance. *Journal of Intelligent Systems* 30, 664–676. http://dx.doi.org/10.1515/jisys-2020-0146.