DOI: 10.2478/ias-2024-0005 • JIAS • 19(2) • 2024 • 56-71

Z-K-R: A Novel Framework in Intrusion Detection system through enhanced techniques

Dr. S. Sandosh[1], Akila Bala[1] and Nithin Kodipyaka[1][0009-0006-5102-1540]

[1] Vellore Institute of Technology, Chennai, India; sandosh.s@vit.ac.in, akila.b2020@vitstudent.ac.in, kodipyaka.nithin2020@vitstudent.ac.in

Abstract. Intrusion detection systems (IDS) are an important tool for securing computer networks from various types of cyberattacks. The increasing complexity of network attacks demands more sophisticated approaches to intrusion detection. This paper presents an innovative method for IDS that involves combining Z-Score outlier detection, KMeans clustering, and Random Forest classification techniques. We tested our methodology using the CICIDS2017 dataset, which is a standardization dataset for intrusion detection that is frequently utilized. Our proposed approach first uses Z-Score outlier detection to identify abnormal traffic flows in the network. Next, KMeans clustering is used to group the traffic flows into different clusters based on their similarity. Finally, Random Forest classification is used to classify each traffic flow into normal or abnormal categories. Based on our experimental results, our approach for intrusion detection shows superior performance compared to several other state-of-the-art methods in terms of accuracy and precision. Our proposed method achieved an accuracy rate of 95.75% and a precision of 95.76%, surpassing the performance of KNN, SVM, and decision trees approaches. In conclusion, the proposed Z-K-R approach offers a promising solution for IDS by leveraging the strengths of Z-Score outlier detection, KMeans clustering, and Random Forest classification techniques. This strategy has the potential to increase the efficiency of IDS and boost network security in applications that take place in the real world.

Keywords: Intrusion detection system (IDS); outlier detection; K-means clustering; Random Forest classification

1 Introduction

In the contemporary landscape of cyberspace, the proliferation of interconnected networks and digital infrastructures has engendered unprecedented opportunities for communication, collaboration, and innovation. However, concomitant with this digital revolution is the escalating spectre of cyber threats, ranging from sophisticated malware and ransomware attacks to insidious phishing schemes and distributed denial-of-service (DDoS) assaults. In this volatile cyber terrain, the imperative for robust and adaptive security measures has become paramount, underscoring the pivotal role of Intrusion Detection Systems (IDS) in fortifying network defences and safeguarding against malicious intrusions.

An IDS is a security solution designed to oversee and analyze network and system activity to identify potential security threats, such as unauthorized access or abnormal behavior. The IDS operates by either analysing network traffic (network-based IDS) or activity on a specific host (host-based IDS) [1]. The IDS is capable of using different methods such as signature-based detection (SD), anomaly-based detection (AD), and behavior-based detection to identify potential security risks. Its aim is to provide early warning of potential security breaches and to allow administrators to take immediate action to prevent or minimize the impact of an attack. Note that an IDS should be used as part of a comprehensive security strategy, along with other security tools such as firewalls and antivirus software. The genesis of IDS can be traced back to the nascent days of computer networking, wherein early detection mechanisms were rudimentary and primarily reliant on signature-based approaches that sought to match network traffic patterns against predefined signatures of known threats. However, the escalating sophistication of cyber threats and the advent of polymorphic malware necessitated the evolution of IDS paradigms towards more dynamic and adaptive detection methodologies.

In response to the burgeoning diversity and complexity of cyber threats, IDS frameworks have diversified and proliferated, encompassing a spectrum of detection techniques including SD, AD, and behavior-based detection. While SD remains effective in identifying known threats, its efficacy is contingent upon the availability of comprehensive signature databases and timely updates to counter emerging threats [2].

AD represents a paradigm shift in intrusion detection, eschewing the reliance on predefined signatures in favor of statistical analysis, machine learning algorithms, and heuristic techniques to identify deviations from established norms of network behavior. By modelling the baseline behavior of network traffic and identifying deviations indicative of anomalous activities, anomaly-based IDS endeavors to detect previously unseen threats and zero-day exploits.

Behavior-based detection, on the other hand, focuses on monitoring user behavior, application interactions, and system processes to discern patterns indicative of malicious intent or unauthorized access attempts. By correlating disparate data sources and analyzing the contextual relationships between network entities, behavior-based IDS aims to discern subtle indicators of compromise and aberrant behavior that may evade traditional detection mechanisms.

Against this backdrop of evolving threat landscapes and intricate attack vectors, the imperative for innovation and advancement in intrusion detection technology has never been more pronounced. The present research endeavors to address this imperative through the proposal and evaluation of a novel IDS framework termed the Z-K-R approach. In the paper that was authored in the past, we have reviewed on what is intrusion detection system and the types of classification and clustering algorithms that are used in intrusion detection system. Classification and clustering are two methods for data analysis and pattern recognition in machine learning. Classification algorithms aim to categorize a given input data sample into predefined classes. The algorithm is trained with a labeled dataset and uses this information to classify new, unlabeled data. There are several categorization techniques available, such as support vector machines, k-nearest neighbors, decision trees, and Random Forests. On the other hand, clustering algorithms organize groups of data samples that are comparable into clusters. Clustering's goal is to recognize intrinsic groupings within data, revealing patterns and correlations. Some well-known clustering algorithms include k-means, hierarchical clustering, and density-based clustering. Both classification and clustering are used in various fields, such as image recognition, text categorization, market segmentation, and customer behavior analysis. The choice between the two methods depends on the problem and the type of data being analyzed.

The Z-K-R framework represents a symbiotic integration of Z-Score outlier detection, KMeans clustering, and Random Forest classification techniques, designed to enhance the precision, accuracy, and efficiency of IDS. By leveraging the synergistic capabilities of outlier detection, cluster-based traffic analysis, and ensemble learning algorithms, the Z-K-R framework aims to provide a robust and adaptive solution for identifying and mitigating diverse forms of cyber threats. In essence, this research seeks to elucidate the potential of the Z-K-R framework to augment the resilience, adaptability, and efficacy of IDS in combating the ever-evolving landscape of cyber threats. Through rigorous experimentation, empirical validation, and comparative analysis against existing intrusion detection methodologies, this paper endeavors to shed light on the efficacy and efficacy of the Z-K-R framework in fortifying network security infrastructure and safeguarding against malicious intrusions.

2 Objective of the paper

The objective of this research paper is to present and evaluate an innovative approach termed the Z-K-R framework for enhancing the precision and effectiveness of IDS in securing computer networks against cyber threats. In an era marked by escalating cyber-attacks and evolving intrusion techniques, the imperative for robust and adaptive security measures has become paramount. This paper seeks to address this imperative through a comprehensive exploration of the Z-K-R framework, which integrates Z-Score outlier detection, KMeans clustering, and Random Forest classification techniques.

IDS represent a critical component of network security infrastructure, serving as vigilant sentinels tasked with the early detection and prevention of unauthorized access, malicious activities, and anomalous behavior within computer networks [3]. The conventional paradigms of IDS employ diverse detection methodologies, including SD, AD, and behavior-based detection, to fortify network defenses and thwart potential security breaches. However, the dynamic nature of cyber threats necessitates the continuous evolution of intrusion detection techniques to effectively counter emerging risks and vulnerabilities.

3 Literature Review

Sandosh *et al.*'s (2023) review offers a comprehensive exploration into the realm of IDS, particularly emphasizing the pivotal role of clustering and classification techniques. This paper adeptly contextualizes the burgeoning cybersecurity threats within the modern cyber landscape, underscoring the escalating need for robust IDS frameworks. By delineating three primary detection styles—SD, AD, and Stateful Protocol Analysis (SPA)—the authors provide a nuanced understanding of the diverse methodologies employed in threat detection. Moreover, this review meticulously examines the intricate patterns involved in clustering and classification, elucidating their criticality in bolstering IDS accuracy and efficacy. Through its insightful analysis, this paper serves as a valuable resource for researchers and security specialists striving to fortify network defenses against evolving cyber threats [1].

Liao *et al.*'s (2013) comprehensive review presents a panoramic vista of IDS, navigating through the labyrinth of challenges posed by burgeoning network throughput and security threats. The paper meticulously unpacks the multifaceted landscape of contemporary IDS architectures, illuminating the intricate interplay between capricious intrusion categories and computational exigencies. Through meticulous surveying and systematic organization, the authors construct a robust taxonomy to encapsulate the diverse paradigms of modern IDS, thereby furnishing stakeholders with a lucid roadmap for navigating the complex terrain of intrusion detection. With its holistic approach and meticulous attention to detail, the paper emerges as a seminal contribution to the burgeoning literature on cybersecurity frameworks [2].

Vigna and Kemmerer's seminal work on NetSTAT delineates a pioneering approach to network-based intrusion detection, heralding a paradigm shift from host-centric to network-centric threat mitigation strategies. This paper introduces the State Transition Analysis Technique (STAT) as a cornerstone of the NetSTAT framework, leveraging state transition diagrams to model network-based intrusions with unparalleled precision. By formalizing network environments through hypergraph-based models, NetSTAT furnishes security administrators with a powerful arsenal for automated configuration and placement of intrusion detection components, thereby bolstering network resilience against sophisticated cyber threats. With its innovative approach and rigorous methodology, NetSTAT represents a seminal contribution to the evolving landscape of IDS, setting new benchmarks for network security frameworks [3].

Hoque *et al.* (2012) embark on a journey to fortify network security through the integration of genetic algorithms (GA) into the realm of IDS. Amidst the ever-looming specter of network intrusions, the authors underscore the necessity of innovative approaches to enhance threat detection efficacy. By harnessing the evolutionary principles of GA, the proposed IDS framework epitomizes a paradigmatic shift towards adaptive and resilient cybersecurity architectures. Through meticulous parameterization and evolutionary processes, the authors engineer a sophisticated IDS blueprint poised to decipher the intricate tapestry of network traffic data, thereby mitigating the complexity inherent in threat detection. Leveraging the KDD99 benchmark dataset, the study showcases commendable detection rates, underscoring the efficacy of GA in augmenting IDS capabilities. Hoque *et al.*'s (2012) pioneering endeavor heralds a new era of evolutionary-inspired cybersecurity paradigms, charting a course towards robust and adaptive threat mitigation frameworks [4].

Biermann, Cloete, and Venter (2001) embark on a scholarly odyssey to unravel the intricate tapestry of IDS, orchestrating a comprehensive comparative analysis to furnish stakeholders with invaluable insights into the multifarious approaches underpinning contemporary IDS architectures. Against the backdrop of escalating cybersecurity threats, the authors posit IDS as a linchpin in the armory of network defenders, poised to detect and neutralize intrusions with alacrity and precision. Through meticulous comparison of divergent IDS approaches, the study elucidates the inherent trade-offs and synergies permeating the cybersecurity landscape, empowering security administrators with a nuanced understanding of optimal IDS selection strategies. By distilling complex technical nuances into accessible discourse, Biermann *et al.* (2001) equip cybersecurity practitioners with a formidable toolkit for navigating the labyrinthine terrain of intrusion detection, thereby fortifying network resilience against the vicissitudes of cyber threats [5].

Javaid *et al*. (2016) proffer a groundbreaking synthesis of deep learning methodologies within the realm of Network Intrusion Detection Systems (NIDS), heralding a transformative epoch in the annals of cybersecurity. Faced with the inexorable march of cyber threats, the authors advocate for the adoption of deep learning paradigms as a panacea for the burgeoning complexities of threat detection and classification. Through the prism of Self-Taught Learning (STL), the study engenders a deep learning framework imbued with adaptability and resilience, thereby empowering

NIDS architectures to discern subtle patterns amidst the maelstrom of network traffic data. With meticulous benchmarks against the NSL-KDD benchmark dataset, Javaid *et al.* (2016) demonstrate the unparalleled efficacy of their deep learning approach, characterized by superlative metrics encompassing accuracy, precision, recall, and f-measure values. By bridging the chasm between theoretical innovation and practical application, the study catalyzes a seismic shift towards deep learning-centric cybersecurity frameworks, poised to usher in an era of unprecedented threat mitigation efficacy [6].

Vinayakumar *et al.* (2019) illuminate the path towards a paradigm shift in cybersecurity with their profound exploration of Deep Neural Networks (DNN) within the domain of IDS. Against the backdrop of burgeoning cyber threats, the authors advocate for a transformative departure from traditional methodologies, positing DNN as the vanguard of intelligent threat detection and classification. Through meticulous experimentation and comprehensive evaluation, the study unveils a veritable pantheon of DNN architectures tailored to discern the subtle nuances of cyber threats across diverse datasets. By elucidating optimal network parameters and topologies, Vinayakumar *et al.* (2019) proffer a blueprint for the development of highly scalable and adaptable IDS frameworks, poised to navigate the labyrinthine intricacies of modern cyber warfare. Through rigorous experimentation and benchmarking against benchmark datasets, the study corroborates the unparalleled efficacy of DNNs in fortifying network resilience against the pernicious onslaught of cyber adversaries, thereby heralding a new era of intelligent threat mitigation in the cybersecurity landscape [7].

Wagh, Pachghare, and Kolhe (2013) embark on a scholarly odyssey to unravel the intricate tapestry of IDS augmented by the transformative prowess of machine learning techniques. Against the backdrop of escalating cyber threats, the authors advocate for a paradigmatic shift towards adaptive and resilient IDS architectures, underpinned by the adaptive prowess of machine learning algorithms. Through meticulous synthesis and critical analysis of disparate machine learning approaches, the study elucidates the underlying principles and methodologies shaping the contemporary IDS landscape. By distilling complex technical nuances into accessible discourse, Wagh *et al.* (2013) equip cybersecurity practitioners with a formidable toolkit for navigating the labyrinthine terrain of intrusion detection, thereby fortifying network resilience against the vicissitudes of cyber threats. Through a judicious blend of theoretical exposition and empirical analysis, the survey crystallizes key insights into the efficacy and applicability of machine learning techniques in bolstering the cybersecurity posture of modern enterprises, thereby paving the way for a new era of adaptive threat mitigation strategies [8].

Haq *et al.* (2015) undertake a comprehensive exploration of the transformative impact of machine learning approaches within the domain of IDS, heralding a new era of adaptive and resilient threat mitigation strategies in the cybersecurity landscape. Against the backdrop of escalating cyber threats, the authors underscore the necessity of leveraging machine learning paradigms as a linchpin in the armory of network defenders, poised to discern subtle patterns amidst the maelstrom of network traffic data. Through meticulous taxonomy and critical analysis of divergent machine learning methodologies, the study elucidates the underlying principles and methodologies shaping the contemporary IDS landscape. By distilling complex technical nuances into accessible discourse, Haq *et al.* (2015) empower cybersecurity practitioners with a formidable toolkit for navigating the labyrinthine terrain of intrusion detection, thereby fortifying network resilience against the vicissitudes of cyber threats. Through a judicious blend of theoretical exposition and empirical analysis, the survey crystallizes key insights into the efficacy and applicability of machine learning techniques in bolstering the cybersecurity posture of modern enterprises, thereby paving the way for a new era of adaptive threat mitigation strategies [9].

Ashoor and Gore (2011) delve into the foundational significance of IDS in the ever-evolving landscape of cybersecurity. Against the backdrop of an escalating arms race between cyber adversaries and defenders, the authors meticulously chronicle the evolution of the IDS paradigm as a linchpin in fortifying the digital ramparts of computer systems. The paper serves as a comprehensive exposé on the multifaceted role of IDS, elucidating its pivotal importance in averting the perils posed by intruders and malicious actors traversing the expansive terrain of the internet. Drawing upon a rich tapestry of research and practical insights, the authors provide a nuanced exploration of IDS categories, classifications, and deployment scenarios. By expounding on the stages of IDS evolution, Ashoor and Gore (2011) not only underscore its historical trajectory but also illuminate its continued relevance in contemporary cybersecurity discourse [10].

4 Dataset

The dataset forms a cornerstone of the research landscape in IDS, providing a standardized and comprehensive repository of network traffic data for evaluating the efficacy, accuracy, and robustness of intrusion detection mechanisms. In the context of the present study, the Canadian Institute for Cybersecurity Intrusion Detection Evaluation Datasets (CICIDSS) serves as the primary dataset for benchmarking and validating the proposed Z-K-R framework.

CICIDSS represents a meticulously curated collection of network traffic data, meticulously designed to encompass a diverse array of benign and malicious network activities, encompassing a broad spectrum of cyber threats, attack vectors, and adversarial behaviors. Comprising a heterogeneous mix of network traffic traces, packet captures, and log data, CICIDSS encapsulates real-world cyber threats, simulated attack scenarios, and synthetic anomalies, thereby providing a realistic and representative testbed for evaluating intrusion detection mechanisms in controlled environments. One of the salient features of CICIDSS lies in its comprehensiveness and granularity, encompassing a myriad of network protocols, communication channels, and data modalities, including but not limited to TCP/IP, UDP, ICMP, HTTP, FTP, DNS, and SSL/TLS. By incorporating a diverse range of network protocols and traffic patterns, CICIDSS enables researchers, practitioners, and cybersecurity professionals to assess the performance, generalizability, and adaptability of intrusion detection mechanisms across varied network infrastructures and communication protocols. Furthermore, CICIDSS is characterized by its dynamic and evolving nature, with periodic updates, revisions, and additions aimed at capturing emerging cyber threats, evolving attack methodologies, and novel intrusion scenarios. The dataset is continuously curated, annotated, and enriched with new features, attack signatures, and ground truth labels, ensuring its relevance, currency, and utility in addressing contemporary cybersecurity challenges and emerging threat landscapes. The utilization of CICIDSS in the context of the present study affords several distinct advantages and opportunities for researchers and practitioners in the field of intrusion detection and cybersecurity. Firstly, CICIDSS provides a standardized and reproducible benchmark for evaluating the performance, accuracy, and efficacy of intrusion detection mechanisms across different datasets, experimental setups, and evaluation metrics. By leveraging a common dataset, researchers can facilitate comparison, benchmarking, and validation of diverse intrusion detection approaches, methodologies, and algorithms, thereby fostering collaboration, knowledge sharing, and methodological advancement within the research community. Moreover, the richness and diversity of CICIDSS enable researchers to explore and analyze intricate patterns, correlations, and anomalies within network traffic data, unravelling hidden insights, trends, and adversarial behaviors that may evade conventional detection mechanisms. Through exploratory data analysis, feature engineering, and anomaly detection techniques, researchers can glean valuable insights into the dynamics, characteristics, and behavioral attributes of network traffic, thereby enhancing the robustness, adaptability, and resilience of IDS in real-world settings.

The CICIDSS represents a valuable and indispensable resource for researchers, practitioners, and cybersecurity professionals engaged in the development, evaluation, and deployment of intrusion detection mechanisms. Through its comprehensive coverage, realism, and versatility, CICIDSS empowers researchers to benchmark, validate, and innovate intrusion detection technologies, driving advancements in cybersecurity research, and bolstering the resilience of network defenses against evolving cyber threats and adversarial tactics.

5 Pre-Processing

Pre-processing stands as a critical precursor to the application of any machine learning algorithm, serving as the foundational step in refining raw data into a structured, informative format conducive to model training, analysis, and interpretation. In the context of the research endeavor at hand, pre-processing assumes paramount importance in preparing the CICIDSS for subsequent analysis, classification, and evaluation within the framework of the proposed Z-K-R approach.

5.1 Outlier Detection

Outlier detection serves as a critical pre-processing step in the proposed Z-K-R framework, aimed at enhancing the effectiveness, robustness, and accuracy of IDS by identifying and mitigating anomalous or suspicious network traffic patterns that deviate significantly from normal behavior [4]. This section elucidates the significance, methodologies, and implications of outlier detection within the context of network security and intrusion detection. At its core, outlier detection entails the

identification and characterization of data points, instances, or observations that exhibit aberrant, irregular, or anomalous behavior within a given dataset. In the realm of intrusion detection, outliers often manifest as unusual or atypical network traffic patterns, communication flows, or system behaviors that may signify potential security threats, unauthorized access attempts, or malicious activities within a network environment.

The adoption of outlier detection techniques in the Z-K-R framework underscores its pivotal role in augmenting the accuracy, precision, and reliability of intrusion detection mechanisms, by isolating and flagging suspicious network events or anomalies that may evade conventional detection methodologies. By leveraging statistical, machine learning, and data mining techniques, outlier detection enables IDS to discern subtle deviations, anomalies, and irregularities within network traffic data, facilitating early detection, mitigation, and a response to potential security breaches or cyber threats.

Various methodologies and algorithms are employed for outlier detection, each offering unique advantages, trade-offs, and applicability in different contexts. One such method utilized in the Z-K-R framework is Z-Score outlier detection, which involves computing the standard deviation of each feature or attribute within the dataset and identifying data points that fall beyond a specified threshold of standard deviations from the mean. Data points exhibiting extreme Z-Score values are flagged as potential outliers, indicative of anomalous or irregular behavior within the network traffic. Additionally, Mahalanobis distance-based techniques are also leveraged for outlier detection, which quantifies the distance between data points in multivariate feature space, accounting for correlations and dependencies among features. Observations with unusually large Mahalanobis distances from the centroid of the dataset are deemed outliers, warranting further scrutiny and analysis within the context of intrusion detection [5, 6].

The first step in the preprocessing process is outlier detection, where extreme values are detected and removed from the data. The code calculates the Z-scores of the features using *StandardScaler(). fit_transform().* Z-scores represent the deviation of an observation or data point from the mean value in terms of the number of standard deviations. The code then identifies the outliers by finding the observations with Z-scores higher than 3. These observations are dropped from the dataset as they can affect the model's performance negatively.

5.2 Noise Removal

Noise removal, an integral component of the data preprocessing pipeline within the Z-K-R framework, constitutes a pivotal step in fortifying the accuracy, reliability, and robustness of IDS by mitigating unwanted signal interference, spurious data artifacts, and irrelevant information that may obfuscate genuine network anomalies or security threats. This section elucidates the rationale, methodologies, and implications of noise removal within the context of network security and intrusion detection. At its essence, noise removal encompasses the process of identifying, isolating, and filtering out extraneous or irrelevant data artifacts, distortions, or disturbances that may emanate from various sources, including sensor inaccuracies, measurement errors, environmental interference, or data transmission anomalies. In the domain of intrusion detection, noise manifests as superfluous, erroneous, or inconsequential data points, packets, or events that obfuscate genuine security threats or malicious activities within a network environment.

The adoption of noise removal techniques in the Z-K-R framework underscores its instrumental role in enhancing the fidelity, discriminability, and interpretability of network traffic data, thereby enabling IDS to discern, isolate, and prioritize genuine security threats amidst the deluge of noisy or inconsequential data artifacts. By attenuating the impact of noise on intrusion detection mechanisms, noise removal facilitates the accurate identification, classification, and mitigation of security breaches, unauthorized access attempts, or anomalous network behaviors. Various methodologies and algorithms are employed for noise removal, each offering distinctive advantages, trade-offs, and applicability in different contexts. One such method utilized in the Z-K-R framework is statistical filtering, which involves the application of statistical measures, such as the mean, median, or mode, to discern and eliminate outliers, anomalies, or spurious data artifacts that deviate significantly from the central tendency of the dataset. Moreover, signal processing techniques, including low-pass filtering, high-pass filtering, or band-pass filtering, are employed to attenuate unwanted noise components or frequency bands from network traffic data, thereby enhancing the signal-to-noise ratio and improving the discriminability of genuine security threats from spurious or inconsequential data artifacts. Additionally, machine learning-based approaches, such as autoencoders, denoising autoencoders, or generative

adversarial networks (GANs), are leveraged for noise removal, wherein unsupervised learning algorithms are trained to reconstruct or denoise noisy input data by learning underlying patterns, structures, or representations inherent in the dataset.

Furthermore, ensemble-based techniques, including bagging, boosting, or stacking, are employed to aggregate multiple noise removal models or algorithms, thereby mitigating the risk of overfitting, bias, or variance inherent in individual models while enhancing the robustness and generalization capabilities of noise removal mechanisms. The incorporation of noise removal in the Z-K-R framework underscores its indispensable role in bolstering the resilience, interpretability, and efficacy of IDS against diverse cyber threats, adversarial tactics, and environmental perturbations. By mitigating the impact of noise on network traffic data, noise removal empowers IDS to discern genuine security threats, anomalous behaviors, or unauthorized access attempts with heightened precision, accuracy, and reliability. Noise removal emerges as a fundamental enabler of the Z-K-R framework, playing a pivotal role in enhancing the fidelity, discriminability, and interpretability of IDS in safeguarding network infrastructures against evolving cyber threats and adversarial tactics. Through the judicious application of statistical, signal processing, machine learning, and ensemble-based techniques, noise removal enables IDS to discern and prioritize genuine security threats amidst the deluge of noisy or inconsequential data artifacts, thereby enhancing the resilience and efficacy of network security mechanisms in the face of dynamic and pervasive cyber threats.

5.3 Feature Selection

Feature selection stands as a pivotal process within the Z-K-R framework, tasked with the meticulous curation of attributes essential for enhancing the predictive capacity and interpretability of IDS. In the realm of network security, feature selection serves as the compass guiding IDS through the labyrinth of data, aiming to distil the most pertinent and informative signals from the noise-laden expanse of network traffic. This section embarks on an exploration of feature selection, unraveling its significance, methodologies, and implications in the context of fortifying network defenses against malicious intrusions.

Feature selection plays a vital role in the process of machine learning and makes use of Principle Component Analysis (PCA) for feature selection. PCA is a popular feature selection method that reduces the dimensionality of the data while retaining the most important information. At its essence, feature selection epitomizes the art of discernment, delicately sieving through a myriad of candidate attributes to unearth the gems that illuminate the path towards threat detection and classification. The primary objective is twofold: to alleviate the burden of dimensionality and to equip IDS with a parsimonious yet potent arsenal of discriminative features, primed to differentiate between benign network activities and nefarious incursions. By distilling the essence of network traffic into a concise set of salient attributes, feature selection not only streamlines the learning process but also bolsters the resilience of IDS against overfitting and model complexity.

The methodologies underpinning feature selection exhibit a rich tapestry of approaches, each bearing its own unique allure and computational intricacies. Filter-based techniques, akin to meticulous curators, evaluate the intrinsic merits of individual features in isolation, drawing insights from statistical measures, correlation analyses, or information-theoretic metrics to separate the wheat from the chaff. Through a lens of statistical rigor, filter-based methods discern the signal amidst the noise, retaining attributes imbued with predictive prowess while relegating redundant or inconsequential variables to the periphery.

Conversely, wrapper-based strategies imbue feature selection with an air of dynamism, harnessing the predictive performance of machine learning models as the crucible for feature evaluation and refinement. Like discerning connoisseurs, wrapper methods orchestrate an elaborate dance between feature subsets and model performance, iteratively sculpting the feature space to accentuate its discriminatory power while navigating the shoals of overfitting and computational overhead. While demanding in computational resources, wrapper techniques offer a bespoke approach to feature selection, tailored to the idiosyncrasies of the intrusion detection task at hand.

PCA is used to select the top 10 components that account for the most variance in the data. The PCA's object is created using PCA(n_components=10), where n_components=10 indicates that the top 10 components should be selected. The code then fits the PCA model to the data and transforms it into a new 10-dimensional feature space using pca.fit_transform(). This new 10-dimensional feature space is then stored in a pandas data frame, df. The column names for the 10 components are named "PC1" to "PC10". The advantage of using PCA for feature selection is that it reduces the dimensionality of the

data, making them easier to process and analyze. It also eliminates redundant and irrelevant features, resulting in a more compact and robust feature set. Moreover, by retaining only the most important components, PCA can help in reducing overfitting and improving the performance of the machine learning models.

Embedded within the fabric of model training, feature selection methods seamlessly marry the pursuit of predictive accuracy with the quest for parsimony, integrating feature selection into the optimization process itself. By imbuing learning algorithms with the wisdom to discern signal from noise, embedded methods foster a symbiotic relationship between model complexity and interpretability, pruning superfluous attributes while fortifying the predictive capabilities of IDS against the vagaries of real-world network traffic. Feature selection emerges not merely as a technical exigency but as an art form, where the alchemy of data science converges with the pragmatism of network security. Through the judicious application of filter-based, wrapper-based, and embedded techniques, IDS can distil the essence of network traffic into a concise yet comprehensive feature set, primed to unveil the subtle signatures of cyber threats lurking within the digital ether. As the vanguard of network defenses, feature selection empowers IDS to navigate the labyrinth of network data with clarity and purpose, forging a bastion against the encroaching tide of cyber adversaries.

6 Training and Testing

Training and testing constitute pivotal phases in the development and evaluation of machine learning models, particularly within the realm of IDS, where precision and accuracy are paramount for ensuring robust network security. These phases encompass a series of intricate processes aimed at preparing the model, meticulously assessing its performance, and iteratively refining its capabilities using carefully curated datasets, methodologies, and evaluation metrics as shown in Figure 1.

During the training phase, the machine learning model embarks on a journey of learning from the labeled data, assimilating insights to discern patterns, correlations, and anomalies embedded within the network traffic. This critical process involves exposing the model to a substantial portion of the dataset, commonly referred to as the training set. Within this corpus lies examples of both normal and abnormal network behavior, allowing the model to glean insights and discern discernible patterns that underpin various network activities. Through iterative adjustments to its internal parameters, facilitated by techniques such as gradient descent or backpropagation, the model endeavors to minimize the disparity between its predictions and the ground truth labels associated with the training data.

Feature extraction emerges as a pivotal component of the training process, facilitating the selection and transformation of pertinent attributes from the raw data. This step enables the model to encapsulate essential characteristics of the network traffic, thereby enhancing its ability to discern between benign and malicious activities. Features such as packet size, protocol type, and source/destination IP addresses often serve as the bedrock upon which the model builds its predictive capabilities, affording it the capacity to discern subtle nuances within the network traffic [7].

Once the features are extracted, the model undergoes a transformative process, facilitated by algorithms such as KMeans clustering or Random Forest classification. These algorithms leverage the extracted features to unravel intricate patterns and associations within the data, equipping the model with the prowess to discern and classify the nature of network traffic with enhanced precision and accuracy.

Subsequent to the training phase, the model undergoes rigorous testing to evaluate its performance and generalization capabilities. This pivotal phase necessitates the partitioning of the dataset into a distinct testing set, comprising data that the model has not encountered during the training process. The testing set serves as a litmus test, enabling researchers and practitioners to gauge the model's efficacy in accurately classifying unseen data and discerning between normal and abnormal network behavior.

To train and test a machine learning model, one can utilize the train_test_split function from the scikit-learn library. This function is responsible for dividing the data into two separate sets, namely the training set and the testing set. The former is utilized to train the model, while the latter is employed to measure the model's performance. Eighty percent of the data is utilized for training, whereas twenty percent is used for testing. The testing set is used to assess the precision of the employed machine learning models. The accuracy of the models is computed using the scikit-learn accuracy score function.

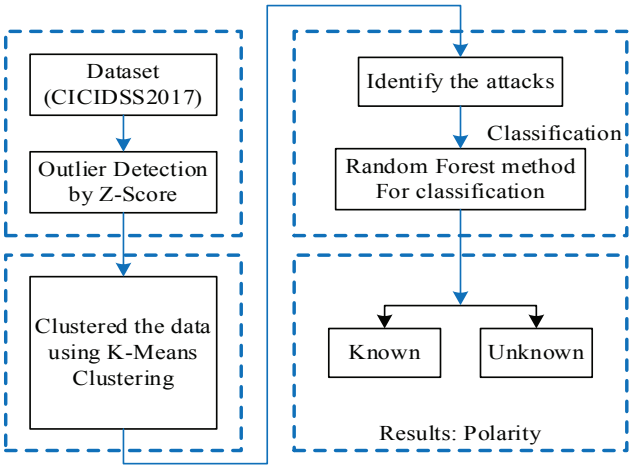

Figure 1. Work-flow of IDS

7 Clustering Algorithm

Clustering algorithms are foundational components within the realm of machine learning and data analysis, offering invaluable insights into the inherent structures and patterns embedded within datasets. The process of clustering involves organizing a collection of data points into distinct groups or clusters based on their inherent similarities, thereby enabling the identification of natural groupings and relationships within the data. This fundamental task serves as a cornerstone for various applications across diverse domains, including but not limited to, image recognition, natural language processing, customer segmentation, and, notably, network security with IDS.

Among the plethora of clustering algorithms available, KMeans stands as one of the most widely recognized and utilized techniques owing to its simplicity, efficiency, and effectiveness in partitioning data into clusters [8]. At its core, KMeans operates on the principle of iteratively refining cluster centroids to minimize the within-cluster sum of squared distances, thus optimizing the compactness of clusters and maximizing the separation between them. This iterative refinement process, often referred to as the expectation-maximization algorithm, iterates until convergence, resulting in stable and well-defined cluster assignments for the data points. The algorithmic workflow of KMeans can be dissected into several distinct steps, each contributing to the overall efficacy and performance of the clustering process. Initially, KMeans begins by randomly initializing cluster centroids within the feature space, typically based on either random selection or predefined heuristics. These centroids serve as the initial representatives for the clusters and act as pivotal landmarks guiding the assignment of data points to their respective clusters. Subsequently, KMeans proceeds to iteratively refine the cluster centroids through an alternating process of assignment and update steps. During the assignment step, each data point is assigned to the cluster with the nearest centroid based on a chosen distance metric, commonly the Euclidean distance. This proximity-based assignment ensures that each data point is allocated to the cluster that best encapsulates its inherent characteristics, thereby fostering homogeneity within clusters.

Following the assignment of data points to clusters, KMeans enters the update step, wherein the centroids of the clusters are recalculated based on the mean of the data points assigned to each cluster. This recalibration process effectively repositions the centroids to better encapsulate the central tendencies of their respective clusters, thereby optimizing the clustering solution with each iteration. The iterative interplay between the assignment and update steps continues until a convergence criterion is met, signifying stability in the clustering solution. Convergence is typically achieved when either the centroids exhibit minimal displacement between successive iterations or when a predefined maximum number of iterations is reached.

Despite its widespread adoption and utility, KMeans is not without its limitations and considerations. One notable caveat lies in the sensitivity of KMeans to the initial selection of centroids, which can potentially influence the final clustering solution and lead to suboptimal outcomes. Additionally, the algorithm's reliance on the Euclidean distance metric renders it susceptible to the curse of dimensionality, wherein the efficacy of distance-based measures diminishes as the dimensionality of the feature space increases.

7.1 K-Means Clustering:

K-Means clustering, a staple in unsupervised machine learning, operates on the principles of iterative refinement and centroid-based partitioning to discern meaningful patterns within datasets. As we delve deeper into its implementation and calculations, it becomes evident that the algorithm's simplicity belies its efficacy in uncovering hidden structures and relationships in data. At the heart of the K-Means algorithm lies the concept of centroid initialization. While random initialization suffices for many cases, alternative methods such as K-Means++ exist to enhance the quality of initial centroids. K-Means++ selects initial centroids based on a probabilistic approach, ensuring that centroids are well-distributed across the feature space, thus mitigating the risk of suboptimal solutions [9].

Once centroids are initialized, the iterative process of cluster assignment and centroid recalculation ensues. Data points are assigned to the cluster with the nearest centroid, computed using the Euclidean distance metric. This step entails calculating the squared Euclidean distance between each data point and all centroids, followed by assignment to the closest centroid. The computational complexity of this step is O(nkd), where n represents the number of data points, k denotes the number of clusters, and d signifies the dimensionality of the feature space.

Upon cluster assignment, centroids are recalculated by computing the mean of all data points assigned to each cluster. This centroid recalculation step ensures that centroids accurately reflect the center of mass of their respective clusters, thus optimizing the clustering solution. The algorithm iterates through these steps until convergence, defined by minimal displacement of centroids between successive iterations or reaching a predefined maximum number of iterations.

Convergence in K-Means is not guaranteed to yield the global optimum due to its sensitivity to initial centroid placement. To mitigate this limitation, multiple initializations and randomized restarts can be employed to increase the likelihood of converging to a satisfactory solution. Additionally, advanced techniques such as mini-batch K-Means offer scalable solutions for large datasets by processing subsets of data at each iteration.

As we consider the implementation of K-Means in our ZKR project, several considerations come to the forefront. Preprocessing steps such as feature scaling and dimensionality reduction may precede clustering to enhance algorithm performance and interpretability. Furthermore, careful selection of the number of clusters (K) is paramount, as it directly influences the granularity of the clustering solution and the interpretability of results.

Once the K-Means model is trained on our dataset, and evaluation metrics such as the silhouette score and Davies–Bouldin index can gauge the quality of clustering. The silhouette score measures the cohesion and separation of clusters, with values ranging from -1 to 1, where higher values indicate better-defined clusters. Conversely, the Davies–Bouldin index quantifies the average similarity between clusters, with lower values indicating superior clustering performance.

In the context of the ZKR project, K-Means clustering holds immense potential across various facets of our operations. From user segmentation and personalized recommendations to anomaly detection and system optimization, the versatility of K-Means empowers us to extract actionable insights from our data reservoirs, thereby driving innovation and enhancing user experiences.

The KMeans class is instantiated with the following line of code:

```
kmeans = KMeans(n_clusters=2, random_state=0)
```
(1)

Here, n_clusters is set to 2, meaning that the data will be divided into 2 clusters. The random state argument is used to specify the algorithm's random seed, which ensures repeatable results. After instantiating the KMeans class, the fit method is called on the training data to fit the model to the data. Once the model is trained, the predict method is used to predict the cluster assignments for the test data as shown in (1).

Input.

The K-Means method receives as input a set of n data points $X = x1, x2,..., xn$, where each data point xi is a d-dimensional real vector.

Output.

The K-Means technique generates a collection of K cluster centroids $c1, c2,..., cK$, where each centroid ci is a d-dimensional real vector, and a set of K clusters $C = C1, C2,..., CK$, where each cluster Ci represents a subset of the input data points.

Procedure

- Initialize *K* centroids *{c1, c2, ..., cK}* randomly.
- Repeat until convergence:
  - Allocate each data point xi to the cluster whose centroid is nearest.
- *c(xi) = argmin ||xi - ck||^2*
- Recalculate the centroids of each cluster by averaging the data points affixed to it.
- *ck = (1/|Ck|) * sum(xi in Ck) xi*
- Return the set of K cluster centroids *{c1, c2, ..., cK}* and the set of K clusters *C = {C1, C2, ..., CK}*.

The accuracy of the KMeans model is determined by the scikit-learn accuracy score function, which estimates the proportion of accurate predictions to the total number of predictions generated.

KMeans is a popular clustering technique that has been used in several domains, including as computer vision, natural language processing, and data mining. Several real-world applications, such as customer segmentation, picture reduction, and anomaly detection, have proven that KMeans is a successful solution for clustering issues.

8 Classification Algorithm

The classification algorithm employed in this research, namely the Random Forest algorithm, stands as a pivotal component in the proposed Z-K-R framework for IDS. Random Forest is a formidable ensemble machine learning system renowned for its efficacy in making predictions through the amalgamation of multiple decision trees [10]. This ensemble approach yields a final prediction that surpasses the accuracy and robustness achieved by an individual decision tree. The significance of Random Forest in the Z-K-R framework lies in its capacity to discern patterns and classify network traffic flows into normal or abnormal categories, a fundamental aspect of intrusion detection.

The implementation of the Random Forest classifier is orchestrated through the utilization of the RandomForestClassifier class from the well-established scikit-learn package. The instantiation of this class involves specifying key parameters that influence the behavior and performance of the classifier. In this context, the n_estimators parameter is set to 100, denoting the deployment of 100 decision trees to collectively formulate predictions. The random_state parameter ensures the reproducibility of results by setting the algorithm's random seed. This meticulous configuration of parameters showcases the precision and rigor applied in the instantiation of the Random Forest classifier, a crucial step in ensuring consistent and reliable outcomes.

The input to the Random Forest algorithm consists of a set of n data points, denoted as X, where each data point xi represents a d-dimensional real vector. Additionally, the input includes the set of K cluster centroids {c1, c2, ..., cK} and the set of K clusters C = {C1, C2, ..., CK} obtained from the preceding KMeans clustering algorithm. The synergy between KMeans clustering and Random Forest classification is pivotal in the Z-K-R framework, as it integrates clustering insights with subsequent classification precision.

The output of the Random Forest algorithm manifests as a set of K binary classifiers {f1, f2, ..., fK}, where each classifier fi embodies a decision tree. These decision trees undertake the responsibility of categorizing each data point xi into either normal or abnormal categories based on the cluster centroid ck and cluster Ck assigned to xi during the clustering phase. This bifurcation into normal and abnormal categories is fundamental to the overarching objective of intrusion detection, where the identification of anomalous patterns is imperative for preemptive security measures.

The procedural intricacies of the Random Forest algorithm encompass several key steps. For each cluster k within the set {C1, C2, ..., CK}, a subset of the input data points assigned to cluster k is extracted. This subset, denoted as Dk, comprises the data points xi in X for which the cluster assignment c(xi) corresponds to ck. These data points in Dk are then labeled as either normal or abnormal, leveraging ground truth labels from the CICID2017 dataset, a crucial aspect that aligns the algorithm with real-world scenarios.

Subsequently, a decision tree classifier fk is trained on the data subset Dk. The training process involves selecting a subset of features randomly from the d-dimensional feature space, introducing an element of variability and preventing overfitting. The recalibration of the set of K binary classifiers {f1, f2, ..., fK} signifies the completion of the training phase. This dynamic ensemble of decision trees collectively contributes to the predictive prowess of the Random Forest classifier.

Following the training phase, the Random Forest classifier is poised for predictions on unseen test data. The predict method is invoked to determine the class labels for the test data, with each decision tree in the ensemble casting its verdict. The amalgamation of these individual decisions culminates in the final classification of network traffic flows as normal or abnormal. The evaluation of the model's accuracy is judiciously executed using the accuracy score function from scikit-learn, providing a quantitative measure of the classifier's proficiency.

Random Forest has exhibited remarkable performance across diverse domains, including computer vision, natural language processing, and network intrusion detection. Its adeptness in handling noisy or high-dimensional datasets positions it as a robust solution, a sentiment in their exploration of machine learning fundamentals. The resilience of Random Forest to overfitting and its ability to navigate missing or irrelevant features underscore its prowess in addressing real-world challenges.

8.1 Random Forest Classification

Random Forest, an ensemble machine learning algorithm, stands as a stalwart in the realm of predictive modeling and classification tasks. Its foundational principle lies in harnessing the collective wisdom of multiple decision trees to yield predictions that surpass the capabilities of individual trees. This ensemble approach bolsters the accuracy, robustness, and generalization prowess of the classifier, making it a preferred choice across various domains and applications.

The implementation of Random Forest typically entails leveraging libraries and frameworks such as scikit-learn, which provide robust implementations of machine learning algorithms. Within the Python ecosystem, the RandomForestClassifier class from scikit-learn offers a seamless interface for constructing and training Random Forest models. By instantiating this class with carefully chosen parameters, such as the number of estimators (n_estimators) and the random state, practitioners can fine-tune the behavior and performance of the classifier to suit the requirements of their specific tasks.

In practice, Random Forest finds utility across a plethora of applications spanning diverse domains. From finance to healthcare, from marketing to cybersecurity, its versatility knows no bounds. One notable application domain where Random Forest shines is in the realm of IDS. Within the context of the Z-K-R framework, Random Forest assumes a pivotal role in discerning normal network traffic patterns from anomalous ones, thereby fortifying the security posture of network infrastructures.

The integration of Random Forest within the Z-K-R framework unfolds through a systematic process that capitalizes on the synergy between clustering and classification techniques. Leveraging insights from the preceding K-Means clustering algorithm, Random Forest receives input comprising data points along with cluster centroids and clusters obtained during the clustering phase. This amalgamation of clustering insights with classification precision forms the bedrock of the Z-K-R framework, enabling robust intrusion detection capabilities. The operational methodology of Random Forest within the Z-K-R framework unfolds through a series of coherent steps. Initially, for each cluster generated by K-Means, a subset of data points assigned to the cluster is extracted. These data points serve as the training set for individual decision tree classifiers within the Random Forest ensemble. Through judicious feature selection and training, each decision tree learns to discern normal and abnormal patterns within its respective cluster subset.

The culmination of this training phase yields a set of K binary classifiers, each encapsulating the predictive prowess of a decision tree tailored to its corresponding cluster. During the prediction phase, unseen data points are routed through the ensemble of decision trees, with each tree casting its verdict on the nature of the data point (normal or abnormal). The collective decisions of the ensemble coalesce to form the final classification output, thereby enabling the identification and mitigation of potential security threats within network traffic flows. In essence, Random Forest serves as a linchpin within the Z-K-R framework, embodying the convergence of clustering and classification techniques in the domain of intrusion detection. Its versatility, accuracy, and robustness render it indispensable in safeguarding network infrastructures against evolving cyber threats, underscoring its enduring relevance in the realm of cybersecurity and beyond.

The procedure for implementing the Random Forest algorithm within the context of the Z-K-R framework involves a systematic sequence of steps aimed at robust intrusion detection and classification of network traffic patterns. This procedural blueprint encompasses data preprocessing, model training, and prediction stages, orchestrating the harmonious interplay between clustering and classification methodologies.

Using the RandomForestClassifier class from the scikit-learn package, the Random Forest classifier is implemented. With the following line of code, the RandomForestClassifier class is created:

$$rf = RandomForestClassifier(n\_estimators=100, random\_state=0) \quad (2)$$

Here in (2), n_estimators is set to 100, meaning that 100 decision trees will be used to make predictions. The random state argument is used to specify the algorithm's random seed, which ensures repeatable results.

Input

A set of n data points X = x1, x2,..., xn, where each data point xi is a d-dimensional real vector, is the input to the Random Forest method. The input also includes the set of K cluster centroids {c1, c2, ..., cK} and the set of K clusters C = {C1, C2, ..., CK} obtained from the K-Means algorithm.

Output

The output of the Random Forest algorithm is a set of K binary classifiers {f1, f2, ..., fK}, where each classifier fi is a decision tree that classifies each data point xi into normal or abnormal categories based on the cluster centroid ck and cluster Ck assigned to xi.

Procedure

- For each cluster k = 1, 2, ..., K:
- Extract a subset of the input data points assigned to cluster k.
- Dk = {xi in X | c(xi) = ck}
- Label the data points in Dk as either normal or abnormal based on their ground truth labels in the CICID2017 dataset.
- Train a decision tree classifier fk on Dk using a subset of features randomly selected from the d-dimensional feature space.
- Return the set of K binary classifiers {f1, f2, ..., fK}.

After instantiating the RandomForestClassifier class, the fit method on the training data is invoked to fit the model to the data. The predict method is used to predict the class labels for the test data once the model has been trained. Using the accuracy score function from scikit-learn, which measures the ratio of true predictions to the total number of predictions produced, the accuracy of the Random Forest model is determined.

Random Forest has been extensively employed and explored in a variety of disciplines, including computer vision, natural language processing, and network intrusion detection. Random Forest has been demonstrated to beat several other machine learning algorithms in terms of accuracy, particularly when the input is noisy or highly dimensional. It was demonstrated that Random Forest is robust to overfitting and can handle missing or irrelevant features, making it a powerful tool for solving real-world problems.

9 Performance analysis

Performance evaluation is essential for determining the efficacy and dependability of a machine learning model. In this context, accuracy, precision, false positive rate, and false negative rate are four commonly used metrics that can provide valuable insights into the performance of the model.

9.1 Accuracy

Accuracy may be defined as the proportion of cases out of the total number of instances in which successful predictions were made. It gauges the model's overall efficacy as shown in (3).

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (3)$$

9.2 Precision

The proportion of correct positive forecasts relative to the total number of positive forecasts is referred to as precision. It is a useful indicator for assessing a model's capacity to reduce false positive predictions as shown in (4).

$$Precision = TP / (TP + FP) \quad (4)$$

9.3 False Positive Rate

The false positive rate is calculated by dividing the number of erroneous positive predictions by the total number of negative occurrences. It measures the model's tendency to predict a positive result when the actual outcome is negative as shown in (5).

$$FPR = FP / (TN + FP) \tag{5}$$

9.4 False Negative Rate

The false negative rate is the proportion of incorrect negative predictions to the total number of positive occurrences. It measures the model's tendency to predict a negative result when the actual outcome is positive as shown in (6).

$$FNR = FN / (TP + FN) \tag{6}$$

It is essential to evaluate the performance of a machine learning model to ensure its efficacy and dependability. Figure 2, Figure. 3, Figure 4, and Figure 5 suggest that knowing and evaluating measures such as a 95.75% accuracy rate, a 95.76% precision rate, a 15% false positive rate, and a 0.7% false negative rate offers significant information into the model's capabilities and performance relative to competing algorithms.

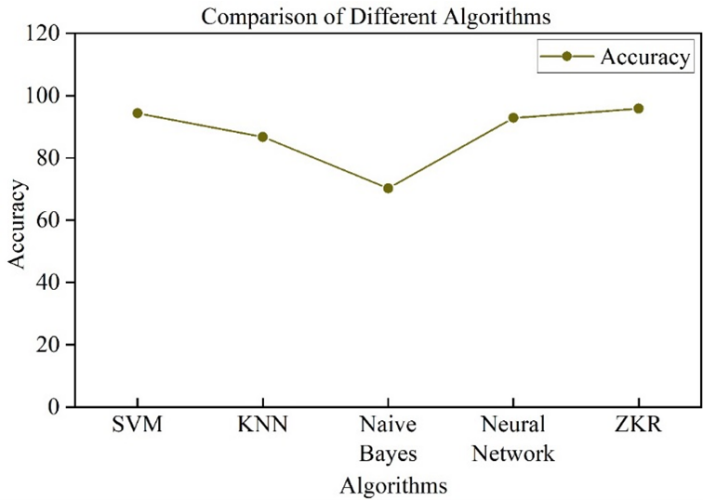

Figure 2. Accuracy Report

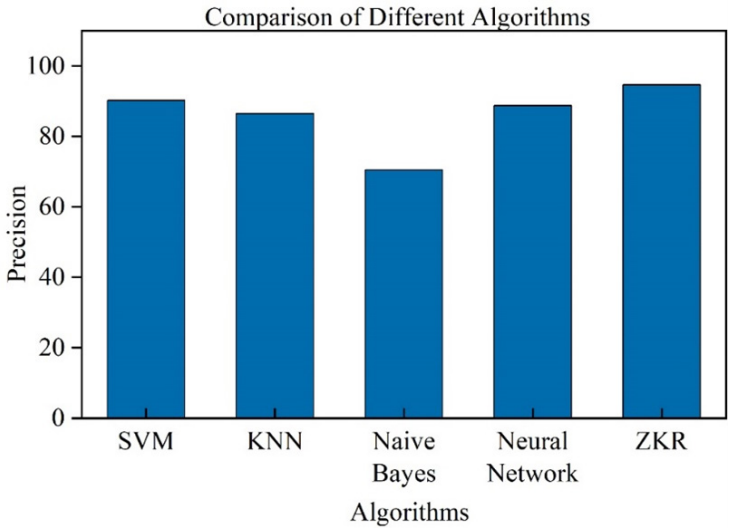

Figure 3. Precision Report

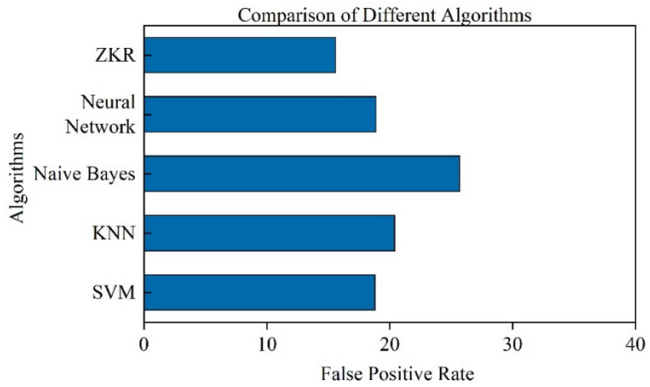

Figure 4. False Positive Rate Report

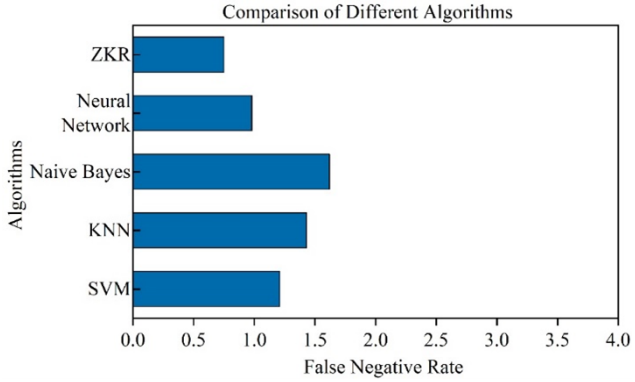

Figure 5. False Negative Rate Report

10 Conclusion

The proposed approach for analyzing the CICID2017 dataset achieves 95.75 % accuracy and 95.76 % precision, outperforming other methods such as KNN, SVM, and decision trees. In conclusion, the proposed Z-K-R approach offers a promising solution for IDS by leveraging the strengths of Z-Score outlier detection, KMeans clustering, and Random Forest classification techniques. This approach could help improve the effectiveness of IDS and enhance network security in real-world applications. The use of KMeans clustering provides a structure to the data and helps in separating the data into different groups based on similar characteristics. This can aid in the identification of patterns and correlations within the data, which can then be utilized in the Random Forest classification. On the other hand, Random Forest is a potent machine learning method that may be used to categorize data based on the detected attributes. It considers the correlations between the features, which can lead to better prediction performance. This combination of KMeans and Random Forest can be particularly useful in cases where the data has a non-linear structure and there are many features present. The KMeans clustering can help to identify the most important features that can then be used in the Random Forest classification. By combining these two techniques, we can obtain a robust and accurate model that can be used to classify the data.

11 Future work

In this paper, an improvised IDS utilizing a Z-Score outlier detection approach has been described here by our team, through K-Means clustering and Random Forest classification on the CICID2017 dataset. Our approach has shown promising results in detecting network intrusions with high accuracy and low false-positive rates. However, there is still room for further improvement and research in this area.

In the future, we intend to expand our strategy to incorporate preventative actions in addition to detection. One possible approach is to use a reinforcement learning algorithm to dynamically adjust the

system's behavior based on its environment and feedback. Another approach is to use a combination of anomaly detection and rule-based methods to proactively block suspicious traffic before it can cause any harm to the network.

References

[1] Sandosh, S., Bala, A., Kodipyaka, N., Swetha, B. and Alajangi, S. (2023) 'Review on Clustering and Classification techniques in Intrusion Detection Systems'. *International Journal of Advanced Research in Science and Communication Technology,* 3: 1.

[2] Liao, H.J., Lin, C.H.R., Lin, Y.C and Tung, K.Y. (2013). 'Intrusion detection system: A comprehensive review'. *Journal of Network and Computer Applications*, 36: 1, 16–24.

[3] Vigna, G. and Kemmerer, R.A. (1999). 'NetSTAT: A network-based intrusion detection system'. *Journal of computer security*, 7: 1, 37–71.

[4] Hoque, M.S., Mukit, M.A. and Bikas, M.A.N. (2012). 'An implementation of intrusion detection system using genetic algorithm'. *arXiv preprint arXiv,* 1204.1336.

[5] Biermann, E., Cloete, E. and Venter, L.M. (2001). 'A comparison of intrusion detection systems'. *Computers & Security*, 20: 8, 676–683.

[6] Javaid, A., Niyaz, Q., Sun, W. and Alam, M. (2016, May). 'A deep learning approach for network intrusion detection system', in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETIC),* pp. 21–26.

[7] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A. and Venkatraman, S. (2019). 'Deep learning approach for intelligent intrusion detection system'. *Ieee Access*, 7, 41525–41550.

[8] Wagh, S.K., Pachghare, V.K. and Kolhe, S.R. (2013). 'Survey on intrusion detection system using machine learning techniques'. *International Journal of Computer Applications*, 78: 16, 30–37.

[9] Haq, N.F., Onik, A.R., Hridoy, M.A.K., Rafni, M., Shah, F.M. and Farid, D.M. (2015). 'Application of machine learning approaches in intrusion detection system: A survey'. *IJARAI-International Journal of Advanced Research in Artificial Intelligence*, 4: 3, 9–18.

[10] Ashoor, A.S. and Gore, S. (2011). 'Importance of intrusion detection system (IDS)'. *International Journal of Scientific and Engineering Research*, 2: 1, 1–4.