Islamic University of Technology (IUT)

Department of Computer Science and Engineering (CSE)

A Feature Selection Approach for Network Intrusion Detections

Authors

A. K. M. Rakinuzzaman - 190041216

Md. Fazle Rabbi Spondon - 190041211

Farhan Fuad - 190041213

Supervisor

Dr. Muhammad Mahbub Alam

Professor, Department of CSE

Islamic University of Technology

Co-Supervisor

S. M. Sabit Bananee

Lecturer, Department of CSE

Islamic University of Technology

Declaration of Authorship

This is to certify that the research done by **A. K. M. Rakinuzzaman, Md. Fazle Rabbi Spondon**, and **Farhan Fuad** under the supervision of **Dr. Muhammad Mahbub Alam**, Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), and **S.M. Sabit Bananee**, Lecturer, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), resulted in the work presented in this thesis. It is further declared that no part of this thesis or the entire thesis has ever been submitted to another institution for a degree or diploma. A list of references is provided, and information taken from both published and unpublished works of others is recognized in the text.

Authors:

A. K. M. Rakinuzzaman
Student ID- 190041216

Md. Fazle Rabbi Spondon
Student ID- 190041211

Farhan Fuad
Student ID- 190041213

Supervisor:

Dr. Muhammad Mahbub Alam
Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Co-supervisor:

S. M. Sabit Bananee
Lecturer
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Acknowledgement

We would like to express our deepest gratitude to **Professor Dr. Muhammad Mahbub Alam** for his exceptional guidance, mentorship, and invaluable contributions throughout our thesis work. As a professor in the department of computer science and engineering at IUT, his extensive knowledge and expertise have been instrumental in shaping our research and academic journey.

From the very beginning stages of our thesis, Professor Dr. Muhammad Mahbub Alam provided us with inspiration and direction. His profound understanding of the subject matter and his ability to offer valuable insights helped us refine our thesis theme and select a suitable subject. His guidance extended to proposing algorithms, making necessary modifications, and providing continuous support throughout the project's implementation and finalization.

We are truly grateful for Professor Dr. Muhammad Mahbub Alam's unwavering commitment to our success. His unwavering dedication to our academic pursuits, his willingness to invest his time and effort in reviewing our work, and his constant availability for discussions and clarifications have been immensely helpful.

Furthermore, we would like to extend our thanks to **S.M. Sabit Bananee**, Lecturer in the Department of Computer Science and Engineering at IUT. His insightful review of our proposal and simulation techniques played a vital role in shaping our research approach. His encouragement and support motivated us to persevere and continue our hard work.

We acknowledge the significant contributions of Professor Dr. Muhammad Mahbub Alam and S.M. Sabit Bananee to our thesis work. Their guidance, advice, and input have been invaluable in ensuring the successful completion of our research. We are truly grateful for their mentorship and are honored to have had the opportunity to learn from them.

Contents

1 Introduction **5**

1.1 Overview 5

1.2 Problem Statement & Contribution 7

1.3 Structure of the Thesis 9

2 Related work **10**

2.1 Overview 10

2.2 Preliminaries 10

2.2.1 Network Intrusion Detection System 11

2.2.2 ML in NIDS 14

2.2.3 Feature Selection 15

2.2.4 Information Theory 16

2.2.5 Mutual Information 18

2.3 Related Work 22

2.3.1 Mutual Information Maximization 22

2.3.2 Mutual Information Feature Selection 23

2.3.3 Minimum-Redundancy Maximum-Relevance 24

2.3.4 Joint Mutual Information 25

2.3.5 DIMRMR 27

2.4 Summary 29

3 Proposed Method **30**

3.1 Motivation 30

3.1.1 Different Techniques of Filter Method 30

3.1.2 Suitability of Mutual Information 31
3.2 Challenges With MI 32
3.2.1 Discretization 33
3.3 Problem Statement 36
3.4 Methodology 36
3.5 Summary 40

4 Experimental Results **41**
4.1 Experimental setup 41
4.2 Data Set 42
4.3 Performance Metrics 44
4.4 Data Preprocessing 45
4.4.1 Discretization 46
4.5 Results 47

5 Conclusion and Future work **53**

List of Figures

2.1 Network Intrusion Detection System (NIDS) 12
2.2 Mutual Information between class and features. 19
2.3 Feature - Class Mutual Information Vs. Feature - Feature Mutual Information 20
2.4 MRMR visualization at 3rd iteration 25
2.5 JMI visualization at 3rd iteration 26
2.6 Results achieved by DIMRMR 28

3.1 JMI_{comp} visualization at 3rd iteration 37
3.2 JMI_{comp} calculation with average 38
3.3 JMI_{comp} comparison between average and minimum 39

4.1 USPS accuracy curve 48
4.2 CNAE-9 accuracy curve 49
4.3 UNSW NB-15 accuracy curve 50
4.4 NSL-KDD accuracy curve 51

Abstract

With the hype that can be seen regarding the application of Artificial intelligence not to places where needed, but rather in places where it is just possible, it is no wonder right now that the approaches in cyber security will also incorporate the utility of Artificial Intelligence. That is already the case for many network intrusion detection system which use many machine learning and deep learning classifiers to sniff out anomaly from a stream of traffics by analyzing different aspects of each network packets. But as the world becomes more digital, the classifiers will have to toil more and more consuming vast resources as their operating necessities a significant amount of computing power to run the complex classifier models. With the help of feature selection methods, the burden on these classifiers can be lessened to the extent of multiple times. Many other endeavors have been undertaken for this purpose. But very few work by keeping the features intact. In doing otherwise, they add another level of obscurity and complexity to the systems. The method proposed in this work is based on the mutual information which provides a minimum subset of them to choose for training the classifier without changing any feature. It is simpler than any dimension reduction technique, yet provides accuracy significantly closer to the far more complex mechanisms. The proposed method is independent of any classifier models and datasets as well, therefore, it can be extended to be used in any supervised learning process lessening the burden of computation without causing much accuracy loss.

Chapter 1

Introduction

Accurate threat detection and strong security in the context of Network Intrusion Detection Systems (NIDS) depend on correctly extracting the relevant information from network traffic. Traditional signature-based NIDS depends on recognizing known harmful patterns, but a more complex approach is necessary given the constantly changing cyberattack scenario. Introducing machine learning-driven feature selection techniques enables NIDS to detect redundant and complimentary features, hence increasing the accuracy of threat detection.

1.1 Overview

One of the biggest challenges in NIDS feature selection is figuring out which characteristics are complementary and redundant. These features effectively add noise and inflate the feature set by carrying information that other features have already gathered. Adding them incurs computational effort and may even obscure the importance of informative traits. Combining these traits with pre-existing ones allows for the discovery of hidden patterns in network activity, which improves the detection of new threats. Accurate and effective threat detection depends on recognizing and utilizing these qualities.

The creation of a condensed and educational feature set is the aim of feature selection in NIDS. It is critical to select characteristics that can reliably differ-

entiate harmful activity from legitimate network data. Removing unnecessary features lowers computational cost and enhances the interpretability of the model. Choosing characteristics that function well together and provide different viewpoints on network behavior is essential for spotting emerging risks. Promising solutions are provided by several feature selection methods that rely on machine learning. The intersection of network security and machine learning is where feature selection in NIDS is located. Using algorithms that are skilled at differentiating between complementarity and redundancy, NIDS can advance beyond straightforward signature-based detection. NIDS may identify known and unknown threats more accurately and efficiently when parsimonious and informative feature sets are built, protecting networks from the ever-changing attacker arsenal. The capacity to uncover the hidden synergy in network traffic data will determine the direction of NIDS as research and new algorithms develop, guaranteeing a future in which threats are found and eliminated before they can penetrate and undermine our vital systems. A more complex approach is needed to handle the ever-changing threat landscape, one that makes use of machine learning-driven feature selection methods. By concentrating on the most informative components of network data, these strategies enable NIDS to not only detect duplicate and complimentary characteristics but also improve threat detection accuracy.

The process of separating complimentary from redundant features in NIDS feature selection is a considerable problem. By definition, redundant features duplicate data that has previously been collected by other features and provide little to no new information. Their addition unnecessarily expands the feature collection, increasing computing costs and perhaps masking the usefulness of meaningful characteristics. Conversely, complimentary characteristics can uncover latent patterns in network activity that are essential for identifying new threats when paired with preexisting ones. Accurate and successful threat detection depends on identifying and making use of these qualities.

Creating a succinct and illuminating feature set that can accurately distinguish between malicious and lawful network traffic is the main objective of feature selec-

tion in NIDS. Removing redundant or unnecessary features not only lessens the computational load but also greatly enhances the interpretability of the final model. To detect new threats, characteristics that complement one another and offer different viewpoints on network behavior must be chosen. Machine learning-based feature selection techniques provide intriguing answers in this regard. By combining techniques from the fields of advanced data analytics and network security, these technologies allow NIDS to go beyond signature-based detection.
In this regard, the very first work was made on maximizing the mutual information. That has been followed by considering redundancies and then joint mutual information. And then including a new item into account, dynamic interaction weight has caused the result to surpass all other techniques in most of the standard datasets.

1.2 Problem Statement & Contribution

Our work aims to solve the following three crucial problems: (1) Feature selection formulas having many terms are relatively less computationally efficient. We aim to utilize fewer terms], which leads to increased computational efficiency; (2) Since feature selection reduces the feature set, the accuracy is always lower than the whole feature set. Therefore, there is room for improvement. We want to introduce a more accurate approximation of the contribution of individual features to classification; and (3) Existing state of the art feature selection method [1] provide an explanation and handle the relationships between newly selected and previously selected features. But we want to successfully capture of the relationships between newly selected and previously selected features which will contribute in better performance. Specifically, we are trying to identify a new feature selection method for NIDS. The need to improve NIDS detection capabilities while maintaining a high degree of accuracy and computing efficiency is what motivates these goals.
We provide a unique feature selection method based on mutual information to overcome these difficulties. A potent statistical metric known as mutual information measures the amount of information gleaned from one random variable through

another. Our method is intended to more efficiently capture the dependence and redundancy information across features than previous approaches by utilizing this measure. The main breakthrough in our method is the identification of a compact formula that makes calculation of the net contribution of a new feature to the classification process easier and enables a more effective feature selection procedure. We have created two enhanced iterations of the feature selection method, building on this basis. These improvements aim to further optimize the selection procedure such that the final feature set is both highly informative and compact, as well as computationally efficient. By concentrating on the connections between characteristics, our strategy ensures that the chosen features make a significant contribution to the classification problem, therefore addressing the shortcomings of conventional approaches. As a result, NIDS performs better overall in identifying recognized as well as unknown dangers.

It can lead to a more efficient and effective feature selection process, which will further the field of NIDS. Our method allows NIDS to reach better levels of accuracy with less processing cost by enhancing the capacity to collect dependency and redundancy information across features. This is especially crucial in the context of contemporary networks, where data volume and complexity are constantly increasing. The future of cybersecurity will be shaped by the capacity to find hidden synergy in network traffic data as NIDS research advances. This will ensure that threats are identified and handled before they have a chance to harm vital systems.

In summary, our suggested mutual information-based feature selection method makes a substantial advancement to the field of NIDS. Our technique presents a viable way to improve threat detection accuracy and reliability in complex network settings by tackling redundancy, complementarity, and computing efficiency issues. This study sets the way for future research targeted at further enhancing the security and resilience of network systems, as well as elevating the state of the art in feature selection for NIDS.

1.3 Structure of the Thesis

The following is the format of the thesis: *Preliminaries* offers a basic understanding of machine learning applications, feature selection strategies, and network intrusion detection systems. The literature on mutual information and related techniques is reviewed under *Related Works. Motivation* talks about the difficulties encountered and the reasoning behind selecting mutual information. The precise research problem being addressed is defined in the *Problem Statement. Methodology* describes the suggested strategy and methods applied. *Obtained Results* summarizes and evaluates the results of the experiments. *Future Work* identifies possible avenues for more investigation. Key contributions and ramifications of the work are outlined in the *Conclusion*.

Summary: This thesis includes preliminaries, related works, motivation, problem statement, methodology, results, future work, and conclusion, outlining our contributions to advancing NIDS capabilities. Our research tackles important issues in Network Intrusion Detection Systems (NIDS) by improving feature selection through a novel mutual information-based technique. Conventional methods struggle with redundant and complementary features, affecting computational efficiency and detection accuracy. We propose a new method that uses mutual information to effectively capture feature dependencies and reduce redundancy, resulting in a more efficient and informative feature set. This method includes a compact formula and two advanced iterations, improving feature selection and overall NIDS performance.

Chapter 2

Related work

2.1 Overview

The availability and integrity of online services are continuously at danger in the ever-changing digital landscape due to malevolent incursions that aim to take advantage of weaknesses and interfere with operations. As vital security measures, Network Intrusion Detection Systems (NIDS) monitor network activity to spot and neutralize such attacks. As sophisticated attacks like Denial-of-Service (DoS) attacks have become more common, NIDS has changed to include cutting-edge methods like machine learning (ML) to improve detection precision and flexibility. An overview of NIDS, its function in cybersecurity, and the incorporation of ML to tackle new issues are given in this chapter. It also presents the fundamental ideas from information theory that underlie these technologies and examines the significance of feature selection in ML model optimization.

2.2 Preliminaries

The foundation for comprehending the main ideas and technologies covered in this chapter is laid forth in this part. The first section provides an overview of Network Intrusion Detection Systems (NIDS), including its features, advantages, and drawbacks in terms of identifying and thwarting threats like denial-of-service (DoS) assaults. After that, the focus switches to how machine learning (ML) can improve NIDS, emphasizing how ML methods increase detection precision and

flexibility. A crucial phase in ML model optimization, feature selection is examined, along with popular approaches like filter, wrapper, and embedding methods. In order to give a theoretical framework for comprehending information measurement and analysis in the context of NIDS and ML, fundamental ideas from information theory are finally introduced, such as entropy, mutual information, and conditional mutual information.

2.2.1 Network Intrusion Detection System

The integrity and availability of online services are always under danger due to hostile intrusions that try to take advantage of weaknesses and interfere with operations in the constantly changing digital world. Network Intrusion Detection Systems (NIDS) operate as watchful sentinels against these enemies, constantly scanning network traffic for unusual activity suggestive of invasions. [1] NIDS functions in a passive manner, examining network traffic that passes through certain locations without interfering with lawful communication. Two basic pillars support their functionality. Network packet data is captured by NIDS, which then extracts relevant data from them, including protocols, payload content, source and destination addresses, and other header fields. For the purpose of identifying unusual patterns or departures from typical network activity, extracted data is examined against known attack signatures or anomaly detection techniques.

Utilizing a wide range of methods, modern NIDS accomplishes successful intrusion detection. Efficient detection of frequently occurring attack types is possible by comparing network traffic patterns to attack signatures that are kept in a central database. Algorithms based on statistics or machine learning detect variations from predetermined benchmarks of typical network activity, which might disclose new or zero-day vulnerabilities. Protocol-specific attacks can be identified and vulnerabilities exploited by attackers can be found via deep analysis of certain protocols and apps. When used properly, NIDS provide a number of benefits in the field of cybersecurity. Early detection of intrusions reduces possible damage by enabling prompt response and mitigation actions. In order to enable forensic

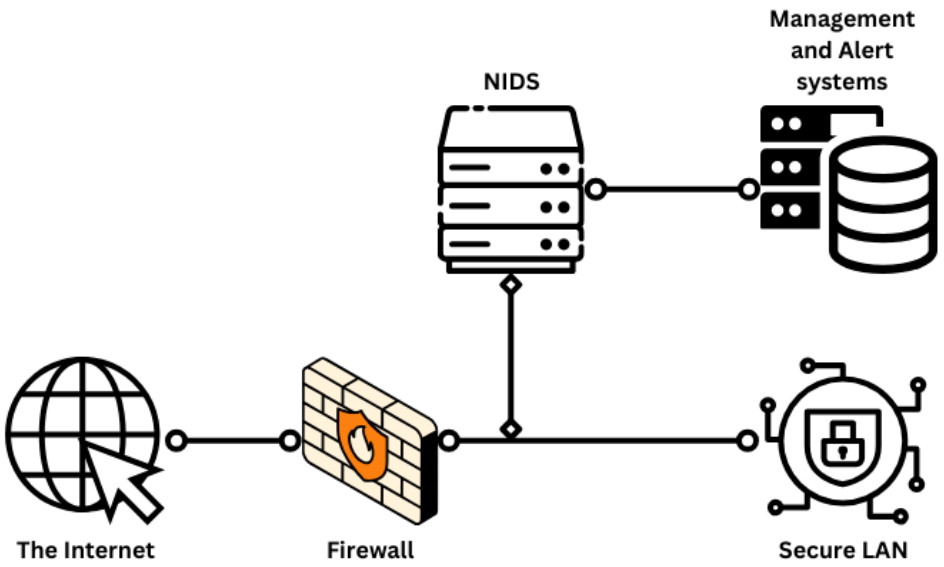

Figure 2.1: Network Intrusion Detection System (NIDS)

analysis and incident response activities, NIDS offers logs and alarms. This makes targeted mitigation and post-event investigation easier. Finding unusual network activity can reveal hidden vulnerabilities and help with patch rollout and preventative actions. NIDS are a useful first line of defense, but they have several drawbacks. Cautious setup and analysis are necessary since anomalous traffic originating from authorized activities may cause false alerts. Complex attackers might use strategies to avoid detection, hence NIDS signatures and algorithms would need to be updated and adjusted on a regular basis. It can require a lot of resources to process high levels of network traffic, which might cause problems for older systems. Network intrusion detection systems are essential for preventing hostile intrusions into online networks and services. N-tier intrusion detection system (NIDS) is an essential line of defense in the intricate world of cybersecurity because it uses cutting-edge detection techniques and can adapt to the ever-changing threat landscape. In the field of cybersecurity, denial-of-service (DoS) attacks provide a distinct problem as they attempt to interfere with online services by flooding them with an excessive amount of malicious traffic. They have a complex connection with Network Intrusion Detection Systems (NIDS) that

calls for a sophisticated knowledge in order to effectively defend against them. DoS attacks work in a number of ways, but their main goal is to overload target systems with traffic, use up resources, and obstruct authorized access. NIDS, which is intended to track network activity and spot abnormal behavior, responds to DoS assaults in a special way. For NIDS, identifying DoS attacks might be difficult for a variety of reasons. Well-planned denial-of-service (DoS) assaults might resemble real-world traffic patterns, hiding their malevolent purpose from algorithms that identify anomalies. Signature-based detection may be hampered by the lack of incorporation of novel DoS attack tactics into signature databases. Large-scale DoS traffic analysis can tax NIDS resources to the limit, making it more difficult for them to find further active incursions. Notwithstanding these difficulties, NIDS is essential for mitigating DoS attacks. Early reaction procedures, which mitigate potential harm and minimize service disruption, are made possible by timely detection of DoS assaults. NIDS-generated logs and alarms offer useful information for forensic investigation, assisting in the identification of attack vectors and vulnerabilities that are exploited. In order to defeat certain DoS techniques, NIDS can be combined with other security technologies to build adaptive mitigation measures like traffic redirection or rate limitation. There are several restrictions on how well NIDS can mitigate DoS attacks. DoS attackers are always coming up with new tricks, therefore NIDS detection capabilities need to be updated and adjusted on a regular basis. Even powerful NIDS may be overwhelmed by large-scale distributed DoS assaults, necessitating a cooperative defense strategy utilizing many network security techniques and protocols. DoS assaults and network intrusion detection systems have a complex connection that is marked by both detection difficulties and significant mitigation possibilities. The successful mitigation of DoS attacks necessitates the integration of NIDS with other security measures and the constant adaptation of detection techniques.

2.2.2 ML in NIDS

Machine learning (ML) has become a powerful tool in modern cybersecurity, supporting Network Intrusion Detection Systems (NIDS) against Denial-of-Service (DoS) assaults. This talk examines how ML and NIDS work together, emphasizing how they improve detection accuracy, allow for more flexible responses, and are always evolving in the face of changing threats. As sentinels in the field of cybersecurity, NIDS keep an eye on network traffic in order to spot malicious activities. With the emergence of ML as a paradigm-shifting technology, NIDS is now able to efficiently identify and mitigate DoS assaults by embracing adaptive, data-driven tactics in place of conventional signature-based strategies. By using ML, network intrusion detection systems (NIDS) may detect DoS assaults proactively by automatically extracting patterns and abnormalities from large datasets of network traffic. Using labeled data as training material, algorithms like Support Vector Machines (SVMs) and Decision Trees are able to categorize network traffic with impressive accuracy, differentiating between DoS assaults and genuine activities. Unusual patterns are found using techniques like clustering and anomaly detection, which reveal new DoS tactics without requiring prior knowledge of attacks. ML-powered network intrusion detection systems are always learning from fresh data, allowing them to adjust to changing threats and stay successful against new attack methods. With improved feature architecture and model optimization, machine learning (ML) improves attack identification while mitigating false positives—a drawback of classic network intrusion detection systems. The impact of DoS assaults may be reduced by using adaptive mitigation techniques like rate limitation, traffic shaping, and intelligent resource allocation, which can be detected and implemented almost instantly thanks to machine learning-driven analysis. The availability of representative, high-quality training and assessment data is critical to the effectiveness of machine learning models. Because ML algorithms might need a lot of resources, real-time implementation within NIDS requires careful optimization. It is still difficult to comprehend how ML models make decisions, which raises questions about possible biases and the necessity of interpreting frameworks. ML

is fundamental to the development of NIDS, transforming their capacity to identify and counteract DoS threats. Cyber security experts can build proactive, adaptable, and durable defenses against these disruptive attacks by leveraging machine learning. A responsible and successful integration of machine learning (ML) inside network intrusion detection systems (NIDS) requires addressing issues with data quality, computational efficiency, and model explainability. It is very possible that more research and development in this area can strengthen cyber security and guarantee the availability and integrity of vital network infrastructure. [2]

2.2.3 Feature Selection

A crucial role that feature selection plays in machine learning (ML) models is to promote increased interpretability, efficiency, and accuracy. This talk goes over the basic ideas of feature selection, discusses its benefits for network intrusion detection systems, and clarifies typical methods used to clean up network data for the best possible intrusion detection. Feature selection becomes an essential method to carefully choose the most significant features, revealing the core of network data and eliminating extraneous noise. ML models show enhanced ability to discern between legal traffic and intrusions by concentrating on the most discriminative aspects. This results in fewer false alarms and higher detection rates. Reducing the feature space frequently leads to faster real-time analysis and model training, which helps NIDS keep up with high-speed network settings and react quickly to threats. By reducing the tendency of machine learning models to overfit to training data, feature selection improves the models' capacity to generalize to new attacks and guarantees reliable performance. A deeper understanding of the network properties that indicate intrusions is facilitated by identifying the most significant aspects, which helps with attack analysis, mitigation techniques, and model explainability.

Common Feature Selection Techniques

1. Filter Methods: These methods, which are independent of the ML model itself, assess features using information-theoretic or statistical metrics. Chi-square

testing, information gain, and correlation analysis are a few examples.

2. Wrapper Methods: By training and assessing machine learning models, wrapper approaches iteratively evaluate feature subsets to find the best feature combination. Recursive feature elimination, forward selection, and reverse elimination are a few examples.

3. Embedded Methods: These strategies, which often use regularization techniques that penalize irrelevant data, directly integrate feature selection into the training process of the machine learning model. Common examples are decision trees with feature significance measures and L1 regularization in linear models.

Understanding network protocols and attack patterns is frequently necessary for effective feature selection in order to inform feature engineering and selection techniques. The specifics of the network traffic data, like its dimensionality, redundancy, and noise levels, determine which feature selection approach is best. Choosing features can require a lot of computing power, especially for complicated machine learning models and huge datasets. It is critical to balance computational economy with accuracy advances. ML-based NIDS rely heavily on feature selection as a revolutionary cornerstone that enables them to identify the most significant patterns within the large volume of network data. Through careful selection of the most relevant elements, NIDS may reach increased levels of precision, effectiveness, readability, and flexibility, enhancing cybersecurity defenses at the front end. [3]

2.2.4 Information Theory

A strong foundation for comprehending and measuring information is offered by information theory. This field is based on fundamental ideas like entropy, mutual information, and conditional mutual information. Every one of these ideas is essential to many different applications, such as machine learning, data compression, and communication systems.

Entropy is a measure of the uncertainty or unpredictability of a random variable. Formally, for a discrete random variable X with a probability mass function *P(X)*, he entropy $H(X)$ is given by:

$$H(X) = -\sum_{x \in X} P(x) \log P(x)$$

where X denotes the set of possible values that X can take. Entropy quantifies the average amount of information required to describe the outcome of X. A higher entropy value indicates greater uncertainty or a more complex distribution.

Mutual Information (MI) measures the amount of information obtained about one random variable through another. For two discrete random variables X and Y with joint probability mass function *P(X, Y)*, mutual information *I(X;Y)* is defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

Conditional Mutual Information (CMI) extends the concept of mutual information to the case where one random variable is conditioned on another. It measures the amount of information shared between two random variables X and Y given a third random variable Z. For discrete random variables, the conditional mutual information the conditional mutual information $I(X;Y \mid Z)$ is given by:

$$I(X;Y \mid Z) = \sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z} P(x,y,z) \log \frac{P(x \mid z)P(y \mid z)}{P(x,y \mid z)}$$

In conclusion, key ideas in information theory include entropy, mutual information, and conditional mutual information. Entropy quantifies the degree of uncertainty in a single variable, mutual information quantifies the degree of information exchanged between two variables, and conditional mutual information goes one step further by taking into consideration other factors that can have an impact on how the primary variables relate to one another. These metrics are fundamental to information analysis and interpretation across a range of scientific and technical domains.

Summary: Network intrusion detection systems, or NIDS, are essential for protecting online services from hostile intrusions because they keep an eye on network traffic and look for patterns or indicators of an assault. They work passively, gathering and deciphering information like addresses, payloads, and protocols. Even though NIDS are useful for early threat detection and incident response, they have drawbacks such as managing false positives and changing attack techniques. ML improves NIDS by using sophisticated data-driven techniques to improve detection accuracy and adaptability. Feature selection, a key process in ML, optimizes model performance by identifying the most relevant features. Information theory concepts such as entropy, mutual information, and conditional mutual information provide essential tools for measuring and interpreting information in this context.

2.2.5 Mutual Information

Between Class and Feature

Regardless of linear or nonlinear correlations, mutual information (MI) is excellent at detecting variables that significantly aid in predicting the class column in a dataset. The MI calculates the bits-per-variable sharing of information between two variables. It expresses the degree to which knowing the value of one variable lessens uncertainty about the other. Determine the MI for each feature-class pair in your dataset by utilizing the relevant libraries or algorithms. Several strategies are used in common estimating. Using joint and marginal probabilities to estimate MI, variables are divided into bins. applying kernel functions to probability density estimation. Greater feature relevance and better links are indicated by higher MI scores. Low MI feature ratings provide less information for class prediction.

$$I(C;Y) = -\sum_{c\in C}\sum_{y\in Y} p(c,y)\log\frac{p(c,y)}{p(c)p(y)}$$

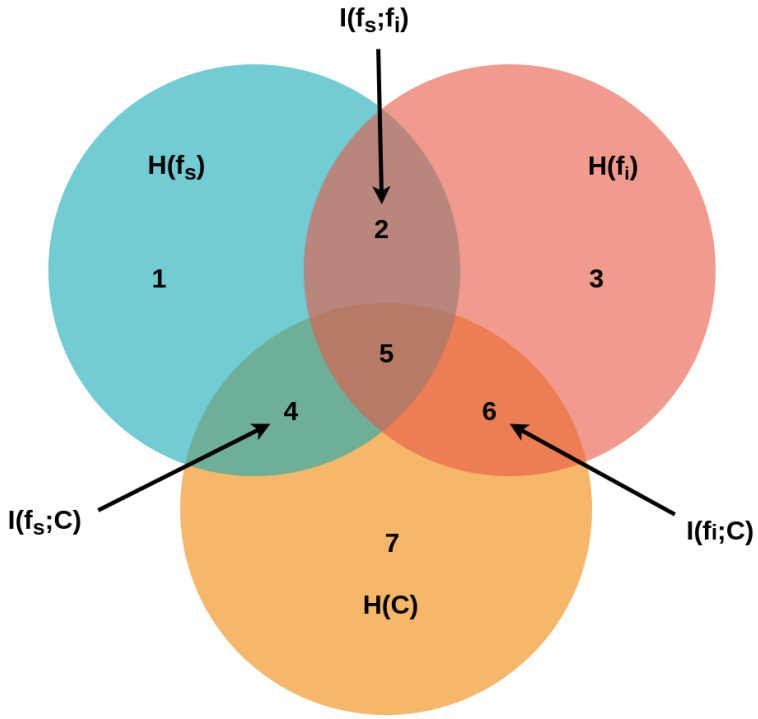

Figure 2.2: Mutual Information between class and features.

Mutual Information Between New Feature and Existing Features

The interaction of a new feature with preexisting features in a dataset can have significant effects on classification performance while searching for the best feature selection. This talk explores the theoretical foundations of feature interactions, clarifies how they affect classification accuracy, and emphasizes how important it is to take feature interactions into account in addition to standard feature relevance metrics for strong model building. Although the relevance of individual features might provide insightful information, features rarely work alone. Analyzing their interactions can reduce overfitting, enhance model generalization, and uncover hidden patterns. Additive effects, in which the combined influence is a linear combination of the separate effects, and complicated relationships, in which the combined effect deviates from simple linear combinations, are two examples of how feature interactions might appear. Including feature interactions can result in decision boundaries that are more thorough, improving model accuracy and flexibility over a range of data distributions. In particular, nonlinear interactions

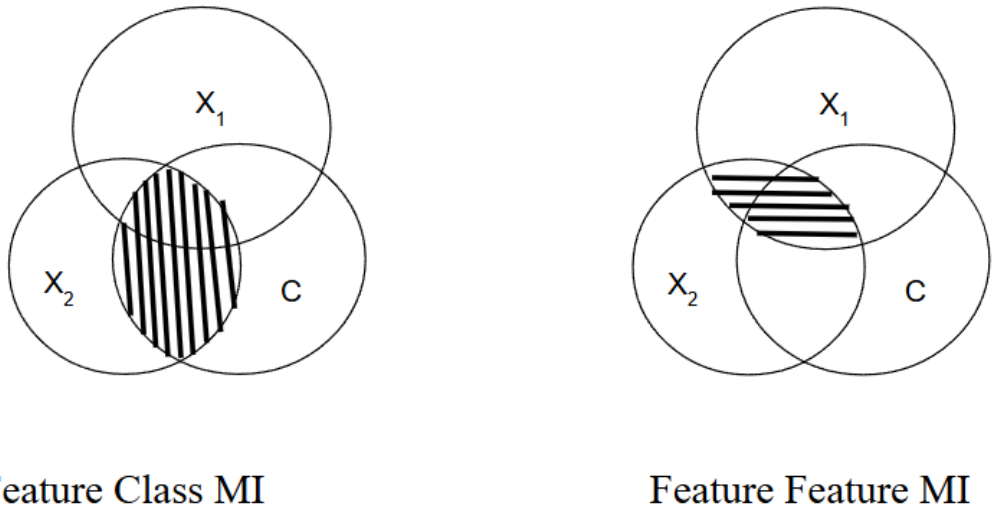

Figure 2.3: Feature - Class Mutual Information Vs. Feature - Feature Mutual Information

can reveal patterns that are hidden from models that just consider the importance of individual features. Models that take feature interactions into account perform better on real-world instances, reduce overfitting, and better generalize to data that hasn't been seen before. While filter methods frequently concentrate on the significance of each individual feature, other feature selection techniques, such wrapper and embedded approaches, naturally take feature interactions into account. To explicitly discover and quantify feature interactions, methods such as random forests, interaction information, and the H-statistic can be used. For improved decision-making, an understanding of feature interactions can provide light on model behavior and make the incorporation of domain information easier. In classification problems, feature interactions are important because they provide a better understanding of how features work together to predict classes. Researchers and practitioners may build more reliable, accurate, and interpretable machine learning models by adopting feature interaction-aware methodologies. This will eventually result in deeper insights and better decision-making across a variety of domains.

Feature Relevant Complementary Item

Although MI illuminates specific relationships, it may miss implicit connections between the new features and current ones. Introducing feature-relevant complementary item (FRCI), a novel idea that explores these cooperative effects in further detail and may be able to get beyond some of the drawbacks associated with focusing only on individual MI ratings.

FRCI measures the additional information about the class that is obtained when a new feature is selected with previous features, in contrast to standard MI, which evaluates the information a new feature gives alone. Essentially, FRCI measures the collective influence of feature combinations, uncovering unnoticed feature contributions that a single MI could overlook. A candidate feature may be dependent or redundant on the earlier selected feature set and it still can have a higher MI value with the target.

The problem of hidden redundancy or dependency is addressed in FRCI's approach. It helps to reward candidate features that increase the total information gain when added to the selected feature set. It reduces the possibility of missing out features which have a better impact when along with the feature set but less impact by itself.

Feature Relevant Complementary Item therefore has the ability to catch complex inter-feature relationships. It has the ability to consider multi-linear relations. These relations goes un-rewarded by feature class MI. Traditional methods tend to overlook these. FRCI helps us to identify such features and reward them in the selection process.

FRCI is to be a powerful tool in the feature selection formula by complementing relations between features. It provides a signified look on feature importance, which was absent in earlier approaches.

Despite having computational overhead and interpretability issues, FRCI holds very potential. It does indeed revolutionize feature selection techniques and help machine learning models to unprecedented levels of accuracy and performance by selecting a smaller feature set that provides better accuracy than other sets of the

same size. [4]

2.3 Related Work

2.3.1 Mutual Information Maximization

The base of feature selection filter algorithms based on information theory is the Mutual Information Maximization (MIM). Although many newer and more performant methods have been inspired from MIM's criterion function, MIM gives us a basic understanding of how features are selected based on their Mutual Information with the target class.

The core idea of MIM is that features with higher levels of mutual information with the class label are more useful for prediction as they contain more information about the class label. [5]

It makes use of the mutual information (MI) to quantify this information match. The features with the greater MI scores with the class are given priority by the MIM criteria function. By calculating the MI between each feature and the class, it selects a smaller feature set among all the features by ordering them in descending order of MI scores.

Because MI is calculated separately for each feature and it has a constant time complexity for every feature, MIM is efficient and can be applied to large-scale datasets.

$$MIM = \arg\max_{fk \in M} \left[I\left(c; f_k\right) \right]$$

Redundancy between features does not impact MIM, which is its main drawback. Features with high individual MI within the class may also be highly linked (redundancy), providing redundant information and therefore making the prediction model less accurate.

Due to its emphasis on individual relevance, MIM may choose elements that are redundant, thus increasing the size of the feature set without benefits. Neglecting

redundancy can lead to less-than-ideal feature sets, which might hamper the accuracy and interpretability of the model.

Mutual Information based feature selection still relies on Mutual Information Maximization as a fundamental idea. Its intrinsic shortcomings in terms of handling feature duplication, however, highlight the significance of utilizing complementary strategies that take feature dependencies and interactions into account. MIM was the initial stepping stone for filter based feature selection methods that use MI. Further improvements has been done on this field that leads us to the present.

2.3.2 Mutual Information Feature Selection

By resolving a significant shortcoming of its forerunner, Mutual Information Maximization (MIM), Mutual Information Feature Selection (MIFS) emerged as a seminal development in information-based feature selection. MIFS takes redundancy into account to create feature sets that are more succinct and informative than MIM, which only considered the significance of each individual feature. However, there may be difficulties with MIFS's dependency on a constant β parameter for redundancy penalization, especially when the number of chosen features rises. This talk examines the theoretical foundations of MIFS, demonstrates its advantages and disadvantages in handling redundancy, and addresses how feature selection techniques may be affected.

$$\text{MIFS} \ = \arg\max_{fk \in M} \left[I\left(c; f_k\right) - \beta \sum_{xk \in F} I\left(f_k; X_k\right) \right]$$

The main advantage of MIFS is its capacity to counteract MIM's drawback of having too many features. This frequently results in more condensed and illuminating feature sets, which may improve the interpretability and performance of the model. But MIFS's fixed β parameter presents a problem. The redundancy penalty term increases with the number of selected characteristics, which may overwhelm relevant concerns and impede the selection of complementary, diversified traits. In order to overcome this constraint, scientists have investigated adaptive β techniques, which dynamically modify the redundancy penalty according on the

available feature set. Aside from mutual information, other methods have looked at conditional mutual information and correlation-based metrics as alternative redundancy measurements. The results of feature selection are greatly influenced by the selection of the β value. Thorough adjustment according to dataset properties and modeling objectives is essential. An important turning point in feature selection is Mutual Information Feature Selection, which highlights the need of taking redundancy into account. Its fixed β parameter, however, needs further study and maybe improvement. Researchers and practitioners can effectively use MIFS to create informative and sparse feature sets by knowing its theoretical underpinnings, strengths, weaknesses, and potential adaptations. However, they must also be aware of its subtleties and consider other approaches to adaptive redundancy management. [6]

2.3.3 Minimum-Redundancy Maximum-Relevance

MRMR(Minimum Redundancy Maximum Relevance) is a method that performs better than the aforementioned MIFS method. MIFS causes fixed penalty for redundancy by tuning the β constant. So this does not perform well always. MRMR replaces this constant with the inverse of selected feature set size. This achieves better performance as it is dynamic based on the selected feature set.

MRMR and similar mutual information-based methods tend to favor features with higher cardinality. This bias must be carefully considered and potentially mitigated. Researchers should be aware of this tendency when applying MRMR or other related techniques to their datasets.

$MRMR = \arg\max_{f_k \in M} \left[I\left(c; f_k\right) - \frac{1}{|F|} \sum_{X_k \in F} I\left(f_k; X_k\right) \right]$

The capacity of MRMR to dynamically control redundancy frequently results in feature sets that are more succinct and informative, improving model interpretability and perhaps lowering processing requirements. Improved Generalization: By reducing duplication and promoting dependence on a variety of complimentary characteristics rather than redundant information, MRMR can enhance model generalization to previously unknown data. The propensity of mutual information-

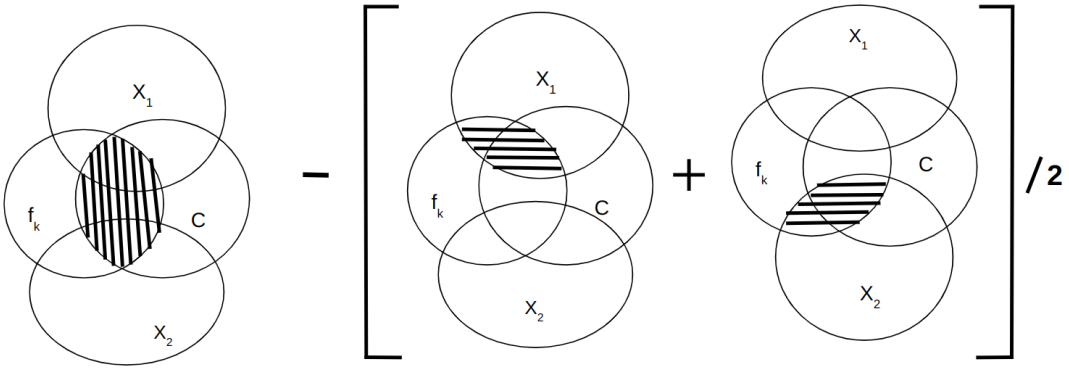

Figure 2.4: MRMR visualization at 3rd iteration

based feature selection methods, such as MRMR, to prioritize features with higher cardinality—that is, more unique values—is a prevalent problem. Lower cardinality informative characteristics may be ignored by this bias. In order to provide a more equitable comparison, researchers have looked at normalization techniques that modify mutual information scores depending on feature cardinality. For a more thorough evaluation of feature significance, complementary feature selection techniques like correlation-based or filter-wrapper hybrid approaches can be used to combat cardinality bias. Minimal Redundancy Max-Relevance solves the fixed redundancy penalty problem of MIFS by providing a more sophisticated method of feature selection. Compact and informative feature sets may be constructed with ease because to its adjustable β parameter. Its cardinality bias must be acknowledged, though, in order to use and understand it correctly. Through an awareness of the theoretical underpinnings, strengths, limits, and potential solutions for mitigation, researchers and practitioners may successfully utilize MRMR's capabilities while guaranteeing impartial and balanced feature selection outputs.

2.3.4 Joint Mutual Information

The Joint Mutual Information (JMI) algorithm was presented by Yang and Moody (1999) as a feature selection method. Joint mutual information (JMI) is a notion that quantifies the mutual dependency between a collection of attributes and the target variable. In order to choose a subset that optimizes the joint information

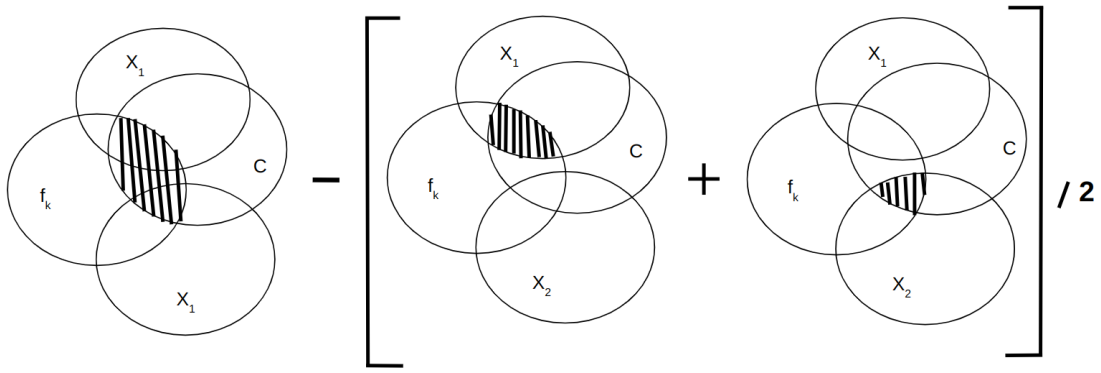

Figure 2.5: JMI visualization at 3rd iteration

shared with the target variable, this strategy takes into account the collective effect of characteristics. The primary benefit of JMI is its capacity to record feature interactions. In contrast to individual MI, which concentrates on individual features, JMI takes into account how characteristics as a whole affect the target variable. With JMI, synergistic linkages that make a major contribution to the prediction job might possibly be captured by choosing feature combinations that display high joint information. By combining the complimentary information provided by several characteristics, classification accuracy may be increased.

$$JMI^{Y\&M} = \arg\max_{f_k \in M} \left\{ I(c; f_k) - \frac{1}{|F|} \sum_{X_k \in F} [I(f_k; X_k) - I(f_k; X_k \mid c)] \right\} \tag{2.1}$$

When JMI is used in place of other feature selection techniques, several research have shown encouraging outcomes. JMI performs better than methods based on individual feature MI. It does better specially on datasets with intricate feature relationships. JMI has some obvious drawbacks. In cases of high-dimensional datasets, computing joint mutual information can be computationally costly, particularly when working with a large number of candidate features and also when the selected feature set becomes fairly large, it will be a time consuming computation for every candidate feature.

JMI has a better potential for feature evaluation because it tries to take into account the whole contribution of the feature set if a candidate feature was to be

selected. JMI is a good candidate where ML application does not introduce the aforementioned drawbacks. But again, noise issue affecting the result is a major drawback of this method. Overall, this method is a classical one and has potential for further improvement. In our thesis we tried to improve the terms from the JMI formula for better computation efficiency and better accuracy.

2.3.5 DIMRMR

Dynamic Interaction-based Minimum-Redundancy Maximum-Relevance (DIMRMR) is the state of the art MI based feature selection method as of the time of this thesis. This method outperforms all other feature selection methods in most of the datasets. DIMRMR achieves better feature set selection by the merit of minimizing the redundancy of the candidate feature with the selected feature set and also maximizing the relevance. With the goal of outperforming existing methods beforehand, this method introduces some special terms in the scoring formula. They are namely C-ratio, Dynamic Interaction weight. This method works on top of the MRMR method's formula and simply introduces the DI multiplier. It considers the feature interaction degree and feature relevant complementary item. Many other methods tend to punish feature complementarity while trying to reduce redundancy. DIMRMR takes this matter into account.

$$DIMRMR = \arg\max_{f_k \in M} \left\{ I(c; f_k) - \frac{1}{|F|} \sum_{X_k \in F} I(f_k; X_k) \right\} * DI(f_k) \tag{2.2}$$

$$DI(f_k) = [1 + C_{ratio}(f_k; F)] * \frac{1}{|F|} \sum_{X_k \in F} I(X_k; c \mid f_k) \tag{2.3}$$

$$C_{ratio}(f_k; F) = 2 \frac{\frac{1}{|F|} \sum_{X_k \in F} I(f_k; c \mid X_k) - I(f_k; c)}{H(f_k) + H(c)} \tag{2.4}$$

On a variety of datasets, experimental findings show that the DIMRMR algorithm delivers optimum classification performance. At 81.52In addition to presenting the idea of feature-relevant complimentary items and using dynamic interaction weight to gauge the significance of candidate features, the study addresses

Datasets	DIMRMR	DRJMIM	IWFS	DWFS	MRMR	RAIW	MRMD	CCMI
Pen recognition	**64.12**	57.41	59.57	59.40	62.62	60.31	61.55	62.77
Ionosphere	85.45	85.36	84.49	83.55	85.42	85.07	**85.76**	85.58
Movement libras	**61.10**	59.18	58.79	59.14	60.37	59.47	60.94	57.98
Mfeat_fac	**88.85**	87.70	87.09	88.15	88.50	88.66	88.08	84.90
USPS	**85.17**	84.92	83.36	79.50	83.12	81.42	82.42	76.38
ORL	**57.17**	50.44	43.43	51.00	55.54	49.24	56.54	52.68
TOX_171	**78.35**	78.35	75.88	76.84	73.17	77.76	73.85	73.12
orlraws10P	**88.49**	73.47	73.02	78.19	87.36	79.40	87.56	84.30
warpPIE10P	90.80	**90.97**	88.14	88.89	90.53	89.14	89.59	85.42
ALLAML	**99.46**	98.94	98.23	99.04	99.05	99.00	98.98	99.11
CLL_SUB_111	**86.51**	86.09	81.04	85.15	85.19	84.53	85.87	85.15
leukemia	**98.87**	97.91	98.43	98.45	98.07	97.92	97.97	97.48
lung_discrete	**82.80**	77.30	74.00	81.20	81.01	80.97	81.42	78.06
GLI_85	**97.46**	97.24	93.46	96.80	95.48	97.23	95.34	95.25
Prostate_GE	95.10	94.72	93.85	95.13	93.85	**95.68**	93.90	94.23
SMK_CAN_187	83.10	**83.16**	82.04	79.64	81.05	80.76	80.81	78.22
arcene	86.63	**87.95**	80.12	82.19	85.06	82.56	83.86	77.99
Avg.	**84.08**	81.83	79.70	81.30	82.67	81.71	82.61	80.51

Figure 2.6: Results achieved by DIMRMR

the standards for differentiating between feature redundancy and dependence in feature selection. Using 10-fold cross-validation on several classifiers, the suggested approach, DIMRMR, is compared with four classical methods and three state-of-the-art methods. A table that displays the mean and standard deviation of the classification accuracy for each dataset as well as the outcomes of various feature selection algorithms is used to compare the classification accuracy on various datasets using a dynamic interaction-based feature selection algorithm. The number of victories, ties, and defeats for each algorithm are also shown in the table. Based on the degree of feature interaction, the DIMRMR algorithm redefines the discriminant criteria of feature dependence and redundancy. When constructing weight, it takes into account the feature-relevant complimentary item for the first time, which improves the algorithm's ability to discriminate between duplication and dependence. DIMRMR is able to distinguish between dependencies and redundancies in features by fusing the dynamic interaction weight with the criteria function of MRMR. With this method, DIMRMR can distinguish between dependent and redundant characteristics with greater accuracy.

2.4 Summary

Dynamic Interaction-based Minimum-Redundancy Maximum-Relevance, or DIMRMR, is a state-of-the-art feature selection method that addresses the shortcomings of previous approaches to improve classification performance. The chapter begins by reviewing popular feature selection methods that balance feature relevance and redundancy, such as Mutual Information Maximization, Min-Redundancy Max-Relevance (MRMR), and Joint Mutual Information. In contrast, DIMRMR distinguishes between feature redundancy and dependency more precisely by combining the idea of feature-relevant complementary items with dynamic interaction weights. By outperforming both traditional and modern approaches on a variety of datasets, this strategy seeks to increase classification accuracy and computing efficiency.

Chapter 3

Proposed Method

3.1 Motivation

3.1.1 Different Techniques of Filter Method

There are four types of filter method in general. These methods are: mutual information, variance threshold, information gain, and correlation-based methods.

1. Correlation-Based Method: These methods quantify the direction and intensity of linear correlations between the target variable and characteristics. They are very good at spotting redundant features and reducing multi-colinearity.

2. Information Gain: This approach measures the decrease in ambiguity surrounding the target variable that results from being aware of a feature's value. It prioritizes characteristics that have a major impact on predicting the target class: uses the uncertainty measure entropy to determine the information gain for every feature.

3. Variance Threshold: This method removes low variance features, which are considered uninformative since they don't vary much between samples: Features with variance below a predetermined threshold are eliminated.

4. Mutual Information: This technique captures both linear and nonlinear interactions by measuring the mutual dependency between two variables:

quantifies the quantity of information gain that one variable offers about another. Possibly displays complicated associations but is more computationally demanding than correlation.

The type of data and the connections between its properties determine how successful filtering techniques are. When compared to wrapper or embedded approaches, filter methods often offer advantages in processing cost and scale well to huge datasets. [7]

3.1.2 Suitability of Mutual Information

Mutual information based filter methods can show well not only the linear but also the non-linear relationships between features and target variable.
The constraints of correlation, which only evaluates linear correlations, are transcended by mutual information. MI measures how dependent two variables are on one another, exposing complex linkages that may be missed using correlation-based techniques. The degree to which knowing the value of a variable lessens uncertainty of another variable is measured by MI. This benefits feature selection. Mutual information can reveal underlying patterns that correlation may overlook in fields like genetics, natural language processing, and cybersecurity where nonlinear interactions are common, resulting in more complete feature sets.
Mutual information extends the range of feature relevance assessment by capturing both linear and nonlinear relations. It is less prone to false correlations that may occur in datasets with a lot of noise. Mutual information can handle non-linear connections well and is robust to monotonic modifications of features, providing flexibility in data preparation.
Mutual Information is also applicable in a range of situations. determining regulatory networks and gene interactions. identifying the semantic connections between concepts and words. examining the relationships between proteins and finding indicators of illness. identifying object connections and texturing patterns. exposing intricate attack techniques and strange network activity.
For big datasets, MI calculation can be computationally intensive, requiring effec-

tive dimensionality reduction techniques or estimate methods. It's important to analyze MI scores carefully since sometimes shared noise or artifacts cause high scores instead of real relationships.

In areas where nonlinear interactions are important, mutual information is a potent tool for feature selection. Researchers and practitioners may increase model performance and interpretability and obtain deeper insights into the underlying structure of their data by embracing its capacity to reveal complex connections. [8]

3.2 Challenges With MI

The measurement of actual contribution towards the class from any new feature becomes computationally difficult and complex with the increasing number of the already selected features. The calculation of conditional mutual information between two random variables given that more than one random variable becomes increasingly difficult and requires multiple nested summation operations resulting it to be a problem of exponential complexity. For example,

$$I(A; B \mid C, D) = \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \sum_{d \in D} P(a, b, c, d) \log \left(\frac{P(a, b \mid c, d)}{P(a \mid c, d) P(b \mid c, d)} \right) \quad (3.1)$$

$$I(A; B \mid C, D, E, F) = \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \sum_{d \in D} \sum_{e \in E} \sum_{f \in F} P(a, b, c, d, e, f) \times \log \left(\frac{P(a, b \mid c, d, e, f)}{P(a \mid c, d, e, f) P(b \mid c, d, e, f)} \right) \quad (3.2)$$

As it can be seen from the above equations, only with 2 selected features, to find the new feature the calculation needs to be performed with 4 nested summations operations. With each addition of a random variable in the conditional part, the equation would require to addition of another nested summation operation.

To avoid these huge computations which will require an unimaginable amount of

time due to its nested attribute, all the techniques regarding mutual information aspire to not calculate exactly, but rather approximate accurately the actual contribution of any new feature to the class compared to already selected features. So far, it has been seen that the most popular approach toward this approximation is by taking the average of some simpler terms.

However average value cannot necessarily represent the actual value of the intended metric. That is why this method always needs to be compensated with some other terms and multipliers. In doing so, more and more terms are added to the approximation equation rendering it polynomially computation-hungry, For example, every addition of any term to the equation causes it to take linearly more time to be computed.

Even after doing all these, it still most of the time becomes not possible to better capture feature and feature relationships properly. Therefore, the selection of features through these processes poses an opportunity to be selected in a more suitable way providing a better approximation. [9]

3.2.1 Discretization

Continuous vs. Discrete Random Variables

Features in a dataset are random variables. A discrete random variable has distinct, countable, finite or countably infinite values. Examples include the number of heads in a coin toss, the number of books checked out from a library, or the number of workers in an office. The probability mass function (PMF) describes the probability distribution of a discrete random variable, where each possible value has a defined probability, and the sum of all probabilities equals 1.

On the other hand, a continuous random variable has values forming a constant range, implying infinitely many values between any two values. Examples include body fat percentage, time taken to run a certain distance, or the weight of objects. The probability density function (PDF) describes the probability distribution of a continuous random variable, with probabilities associated with intervals of values rather than single values, and the total area under the curve of the PDF equals

1. It is important to note that the probability of a continuous random variable taking on a specific value is zero, and probabilities are defined for ranges of values, not individual values.

Key Differences:

- **Countability**: Discrete variables have countable values, while continuous variables have uncountable infinite values.
- **Measurement**: Discrete variables are often counts of items, whereas continuous variables represent measurements along a continuum.
- **Probability Distribution**: Discrete variables use a Probability Mass Function (PMF), while continuous variables use a Probability Density Function (PDF).

CAIM discretization [10]

Discretization plays a crucial role in computing mutual information (MI) because it simplifies the complexity of the data by reducing continuous variables into discrete bins. This simplification makes it easier to calculate the dependency between variables, which is exactly what mutual information aims to measure. Mutual information quantifies the amount of information that knowing one variable provides about another. In the context of machine learning and statistics, higher mutual information indicates a stronger dependency between variables.

Calculating mutual information directly becomes computationally expensive. Because the joint probability density function of the variables needs to be integrated over. If the variables are discrete instead, the integration is no longer required. The computation then only requires summation operations. Thus allowing simpler and faster computation.

Let us assume two continuous random variables X and Y, and their corresponding discrete versions as X_d and Y_d, respectively.

The mutual information between X and Y is defined as:

$$I(X;Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P_{XY}(x,y) \log \left(\frac{P_{XY}(x,y)}{P_X(x) P_Y(y)} \right) dxdy$$

Where, P_{XY} is the joint PDF of X and Yrespectively,
P_X and P_Y are the marginal PDF of X and Y respectively.

By discretization, we divide the ranges of X and Y into a number of bins, resulting in discrete variables X_d and Y_d. The mutual information between X_d and Y_d can be approximated as:

$$I(X_d;Y_d) \approx \sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} P_{X_dY_d}(i,j) \log \left(\frac{P_{X_dY_d}(i,j)}{P_{X_d}(i) P_{Y_d}(j)} \right)$$

Where,
N_X and N_Y are the numbers of bins for X and Y respectively,
$P_{X_dY_d}(i,j)$ is the joint PMF of X_d and Y_d,
$P_{X_d}(i)$ and $P_{Y_d}(j)$ are the marginal PMF of X_d and Y_d respectively.

This approximation significantly reduces the computational complexity involved in calculating mutual information, making it feasible to compute even for high-dimensional data. Furthermore, it allows for the application of mutual information in various domains, including feature selection, clustering, and dimensionality reduction, where understanding the dependencies between variables is key to achieving good performance.

Machine learning techniques are frequently used to extract information from databases. Most of these methods are limited to using with data that has discrete numerical or nominal properties (features) to describe it. A discretization algorithm is required in the case of continuous attributes in order to convert them into discrete ones. The class-attribute interdependence maximization algorithm, or CAIM, is described in this study. It is intended to be used with supervised data. Maximizing the class-attribute dependency and producing a (possible) minimal number of discrete intervals are the two main objectives of the CAIM method. Unlike some other discretization methods, this one does not require the user to choose the number of intervals in advance.

Tests conducted with six additional cutting-edge discretization algorithms in addition to CAIM reveal that discrete attributes produced by the CAIM method nearly invariably have the highest class-attribute interdependence and the fewest intervals [10].

3.3 Problem Statement

Any new feature selection method with the potential to perform better than the existing ones would have to overcome its limitations of them. Our problem is to find a new feature selection technique that

- Consists of fewer terms, which results in computational efficiency
- Better approximate the contribution of features toward classification
- Better capture the relationships between new features and selected features

3.4 Methodology

The proposed feature selection method, the first step, consists of fewer terms than the technique known as JMI. We have based our work on this technique because of its better approximation of the actual contribution of features toward classification. This method is relatively more accurate in determining the amount of information added by a single feature to the class. But it is comprised of 3 terms, 2 of which are supposed to be working under the summation operation, requiring the execution of a loop while being computed through any modern programming logic. Therefore, even a simple add or subtract operation would result in a huge amount of computation due to the iterative process it needs to undergo. It will be of great convenience if the number of operations can be decreased, which takes place under the summation operator. Therefore, before proposing any alternate method or different mechanism, there is a necessity to improve the very technique we are starting our work from.

This opportunity has been taken advantage of. Upon analyzing the existing JMI equation, we have observed that it attempts to remove the redundant part brought by any new feature for the class. It does so by approximating the common region between any feature and already existing another feature with respect to the class. Then it takes the average of this number and subtracts it from the mutual information between the new feature and class. As a result it provides us the average of the mutual information between the new feature and class with respect to the previous feature. But to approximate this number, we don't need to calculate separately the common region between any feature and another already existing feature with respect to the class and then subtract it. We can directly calculate the result using a single term under the summation operation as depicted in the following figure.

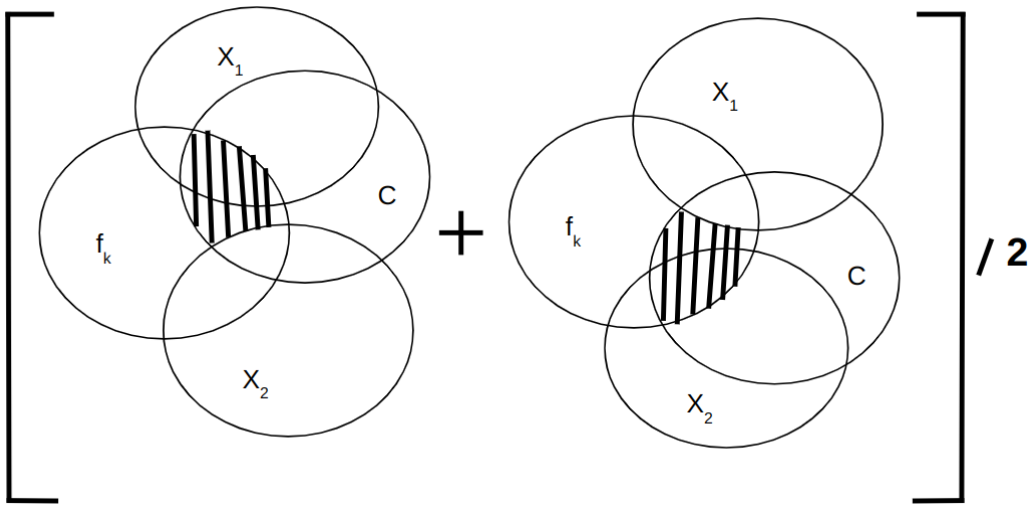

Figure 3.1: JMI_{comp} visualization at 3rd iteration

First, we have decreased the number of terms in our equation than JMI. Where JMI has 3 terms, we have used only one to reach the same prediction as JMI but with better computational efficiency.

$$JMI_{Comp} = \arg\max_{f_k \in M} \left[\frac{1}{|F|} \sum_{X_k \in F} I(f_k; c|X_k) \right] \tag{3.3}$$

For any good approximation, our goal is to calculate the amount of redundant information brought by any new feature with respect to already selected features

and subtract it from the actual mutual information between class and the new candidate feature. To do so, we first calculate the average information commonality between the new feature and already existing features by considering them one by one. It even though gives us a good idea of the redundant information, will fail in case of a scenario where the amount of information shared by the new feature and existing feature varies. As depicted in the following picture, we can see that if a selected feature shares more information with the new feature and other one shares less, then the average of these two numbers would result in lesser number than the bigger one providing us an amount as redundant information which is surely significantly lesser than the actual one.

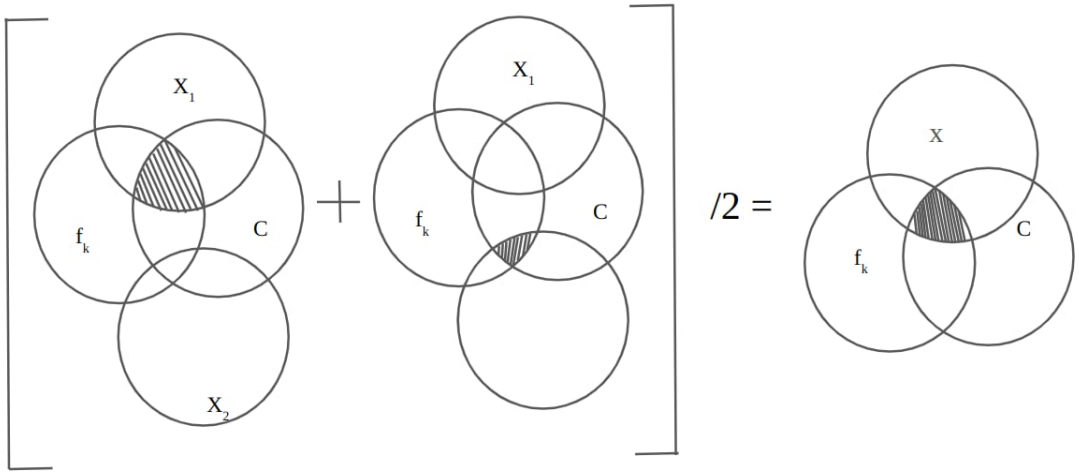

Figure 3.2: JMI_{comp} calculation with average

Our proposed method overcomes this issue. Here, our main goal is to approximate the amount of information provided by the new feature as much accurately as possible. Therefore, instead of taking the average of our compact equation proposed before, we must take the minimum of it. Because, if any of the existing feature has larger common region with our new feature, it will surely be reduced by taking the average as depicted in the figure below. But to have a more accurate idea of the redundant information, we must take it into account. To do so, we have to consider the minimum region shared by our new feature and class with respect to the already existing feature. Only then we will be able to take into account the bigger number as the amount of redundant information, as all other alternate

ways would result in a smaller approximation. As depicted in the following figure, we can see that taking the minimum would provide us with a larger subtraction of the redundant information, resulting in more accurate approximation of the amount of information brought by our new feature.

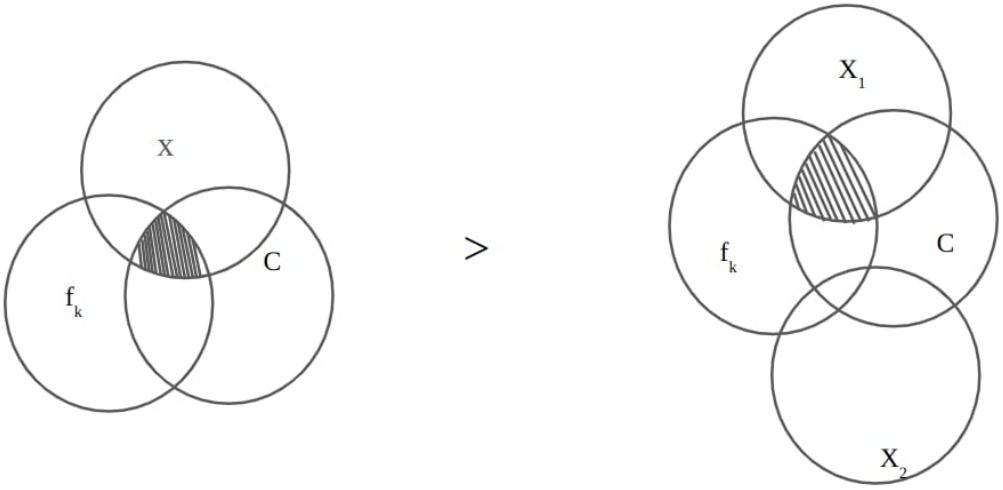

Figure 3.3: JMI_{comp} comparison between average and minimum

So, to have a better approximation than JMI, we have overcome the cliche of using the average, Rather we took the minimum value of the conditional mutual information between the new feature and the class given already selected features providing us closer to the actual contribution of the new feature by disregarding the maximum overlap between the new feature and already selected features.

$$JMI_{Comp\_Min} = \arg\max_{f_k \in M} \left[\min_{X_k \in F} I(f_k; c|X_k) \right] \tag{3.4}$$

After that to better capture the relationship between new features and already existing feature set, a complementary information has been incorporated. The complementary information helps determining the proper contribution of a feature by further trimming the feature score.

$$JMI_{Comp\_CI} = argmax_{f_k \in M}[argmin_{x_k \in F}[I(f_k; c|X_k)] - CI] \tag{3.5}$$

$$\text{where, } CI = argmax_{x_k \in F}[|I(X_k; c|f_k) - I(X_k; c)|]$$

This concept of complementary information or feature relevant complimentary mutual information to be exact has been introduced in the DIMRMR equation. It enable to maximization of relevance while selecting a new feature as well as minimizing its redundancy. Actually it helps calculate the amount of information complemented by the new feature, not only its commonality with the existing feature set. This term is important because the part of complementary information is sometimes punished by other methods while solely trying to minimize the amount of redundant information. However, in the DIMRMR technique, the calculation requires some more terms and at the end a multiplication operation as well. Another item called dynamic interaction weight has been introduced there to perform the multiplication operation with. But for our technique, no extra calculation is needed with the complimentary information, nor is it required to perform any mulplication operation to incorporate it within our equation. A simple subtraction is sufficient to provide a better approximation taking into account the dependency relationship between features along with redundancy as well.

3.5 Summary

In this part, feature selection strategies are discussed with a focus on mutual information (MI), which is important because it may capture both linear and nonlinear correlations between target variables and features. Though it has computational hurdles with growing feature dimensions, MI is resilient to noise and shows complicated connections, unlike correlation-based approaches. Discreteization techniques like as CAIM improve efficiency by reducing the complexity of MI computations by transforming continuous variables into discrete bins. By lowering computing cost, employing minimal conditional mutual information for improved approximation, and adding complementary information to enhance feature relevance and associations, the suggested methodology outperforms current methods.

Chapter 4

Experimental Results

4.1 Experimental setup

The system used for running the experiments is a Core i5-8500k desktop computer having 16 GB of memory and Ubuntu 20.04 LTS operating system. The datasets with continuous variables were discretized with CAIM [10] discretization technique which is a supervised discretization algorithm. All the experiments were done in python. A pip library, Caimcaim[1] was used for the discretization task. The library was further modified for multi-threading[2]. For rest of the experiment tasks such as computing the feature selection method values, calculating accuracy scores, the libraries in use are: Pandas[3], Numpy[4] and Scikit-learn[5].

The generalizability of the model was carefully evaluated using ten-fold cross-validation. An ensemble of 200 decision trees was employed by the Random Forest, which was set up using a Gini impurity criterion, restricted tree depth (5), and square root feature selection, to improve its predictive power. [11]

[1]https://github.com/airysen/caimcaim
[2]https://github.com/RakinRkz/caimcaim
[3]https://pandas.pydata.org
[4]https://numpy.org
[5]https://scikit-learn.org

4.2 Data Set

The following section provides an overview of the datasets used in this study. These datasets, namely CNAE-9, USPS, UNSW-NB15, and NSL-KDD, span diverse domains such as text classification, image recognition, and network intrusion detection. Each dataset is described in details, highlighting its features, target variables, and relevance to the research.

1. CNAE-9 Dataset

- **Description**: The CNAE-9 dataset is a text classification dataset that contains descriptions of Brazilian companies categorized into 9 distinct categories based on their economic activities. Each instance in the dataset represents a company, and the features are word frequencies derived from the company's description.
- **Features**: The dataset consists of 1,080 instances and 856 features, where each feature corresponds to the frequency of a specific word in the company's description.
- **Target Variable**: The target variable is a multi-class label representing one of the 9 categories.
- **Use Case**: This dataset is commonly used for text classification and feature selection tasks in machine learning.

2. USPS Dataset

- **Description**: The USPS (United States Postal Service) dataset is a collection of handwritten digit images used for digit recognition tasks. It contains grayscale images of digits (0–9) scanned from envelopes by the U.S. Postal Service.

- **Features**: Each image is of size 16x16 pixels, resulting in 256 features per instance. The pixel values are normalized to a range of 0 to 1.
- **Target Variable**: The target variable is a multi-class label representing the digit (0–9) in the image.
- **Use Case**: This dataset is widely used for benchmarking classification algorithms, particularly in the domain of optical character recognition (OCR) and image processing.

3. UNSW-NB15 Dataset

- **Description**: The UNSW-NB15 dataset is a network intrusion detection dataset created by the Australian Centre for Cyber Security (ACCS). It contains a hybrid of real modern normal activities and synthetic attack behaviors.
- **Features**: The dataset consists of 49 features, including flow-based features (e.g., duration, protocol, packets) and content-based features (e.g., HTTP methods, DNS queries). It contains over 2.5 million instances.
- **Target Variable**: The target variable is binary, indicating whether the instance is normal or an attack. It also includes multi-class labels for specific attack types (e.g., Fuzzers, Analysis, Backdoors, DoS, Exploits, etc.).
- **Use Case**: This dataset is widely used for developing and evaluating intrusion detection systems (IDS) and anomaly detection algorithms.

4. NSL-KDD Dataset

- **Description**: The NSL-KDD dataset is an improved version of the KDD Cup 1999 dataset, designed for network intrusion detection. It addresses some of the issues in the original KDD Cup 1999 dataset, such as redundancy and imbalance.

- **Features**: The dataset contains 41 features, including basic connection features (e.g., duration, protocol type), content features (e.g., number of failed login attempts), and traffic features (e.g., count of connections to the same host).
- **Target Variable**: The target variable is binary, indicating whether the instance is normal or an attack. It also includes multi-class labels for specific attack types (e.g., DoS, R2L, U2R, Probe).
- **Use Case**: This dataset is widely used for evaluating intrusion detection systems (IDS) and machine learning models for cybersecurity.

Dataset Summary

Datasets	Samples	Features	Class types
USPS	9298	256	10
CNAE 9	1080	856	9
UNSW NB15	2540044	49	9
NSL KDD	148517	41	4

Table 4.1: Dataset Summary

4.3 Performance Metrics

To evaluate the performance of each feature selection method, we employed **10-fold cross-validation accuracy** as the primary metric. This allows for a fair comparison of the methods by assessing how each set of selected features influences the model's predictive performance.

The feature selection methods in comparison are used to select a number of features from the datasets, giving us a priority list of top features selected by every method. The accuracy score is then acquired for the priority list, with increasing the allowed number of features by 1 upto one-third of the total feature count of the dataset if the total number of features is less than or near 50. Otherwise, if

the data set has a large number of features, we go up to 50 features and perform the tests.

10-Fold Cross-Validation Accuracy

10-fold cross-validation involves partitioning the dataset into ten equal-sized subsets, known as *folds*. The model is trained on nine folds and tested on the remaining fold. This process is repeated ten times, with each fold serving as the test set once. The accuracy from each iteration is recorded, and the overall performance is determined by averaging these accuracy scores. This technique helps mitigate issues like overfitting and provides a more generalized evaluation of the model's performance.

Implementation Details

- **Data Partitioning:** The dataset was randomly shuffled and divided into ten equal parts, ensuring each fold is representative of the overall data distribution.
- **Model Training and Evaluation:** For each fold, the model was trained on the combined data of nine folds and evaluated on the remaining fold.
- **Performance Aggregation:** The accuracy scores from all ten folds were averaged to obtain a comprehensive measure of the model's predictive accuracy.

By utilizing 10-fold cross-validation accuracy, we ensure a reliable and unbiased evaluation of our model, accounting for variability within the data and enhancing the credibility of our performance assessment.

4.4 Data Preprocessing

Mutual information (MI), which measures statistical relationships between variables, is difficult to compute directly for continuous data. By converting continuous

variables into discrete representations, discretization fills this gap and makes MI estimation useful and understandable.

The estimation of probability density functions (PDFs), which is necessary for continuous MI, is intrinsically non-parametric and data-intensive. Density estimation produces biased or unstable MI values in finite datasets, particularly when there are few samples or high dimensionality. Discretization gets around this by using binning to transform PDFs into probability mass functions (PMFs), which reduces estimate to frequency counting, a method that is both computationally feasible and statistically sound.

Discreteization also harmonizes different kinds of data. There is no standard framework for MI computations between heterogeneous variables, yet many real-world datasets combine continuous and categorical properties. Because discretization guarantees homogeneity, MI may be calculated uniformly across all features.

4.4.1 Discretization

Discretization techniques can be broadly classified into supervised and unsupervised methods. Unsupervised methods, such as equal-width and equal-frequency binning, partition the data based on predefined criteria without considering class labels. In contrast, supervised discretization methods leverage class information to optimize the partitioning process, thereby improving classification accuracy.

Class-Attribute Interdependence Maximization (CAIM)

The CAIM (Class-Attribute Interdependence Maximization) algorithm is a supervised discretization method specifically designed for classification tasks. It aims to maximize the dependency between discretized attribute intervals and class labels, ensuring that the resulting intervals are meaningful and contribute effectively to classification accuracy. By using CAIM, we achieve:

- **Maximization of Class-Attribute Dependency:** CAIM optimizes the discretization process by selecting intervals that maximize the interdepen-

dence between attributes and class labels. This results in a more informative and discriminative feature space.

- **Reduction of Data Dimensionality:** By converting continuous attributes into a minimal number of discrete intervals, CAIM simplifies the dataset while preserving essential information, reducing computational overhead.
- **Enhanced Classification Performance:** The algorithm ensures that each interval contains instances that predominantly belong to a single class, improving classifier performance by minimizing misclassification errors.
- **Automatic Determination of Intervals:** Unlike traditional binning methods that require predefined parameters, CAIM dynamically determines the optimal number and limits of intervals based on the data distribution.

4.5 Results

Our methods were first applied on the generic datasets to recreate the literature and ensure that our method works better than the existing methods. We first present results on two datasets, USPS and CNAE-9. We can see the usual behavior in the accuracy curve among MIM, MRMR and DIMRMR. DIMRMR is the most performant among the three. And our method performs slightly better than or as good as DIMRMR as seen in the accuracy curves.

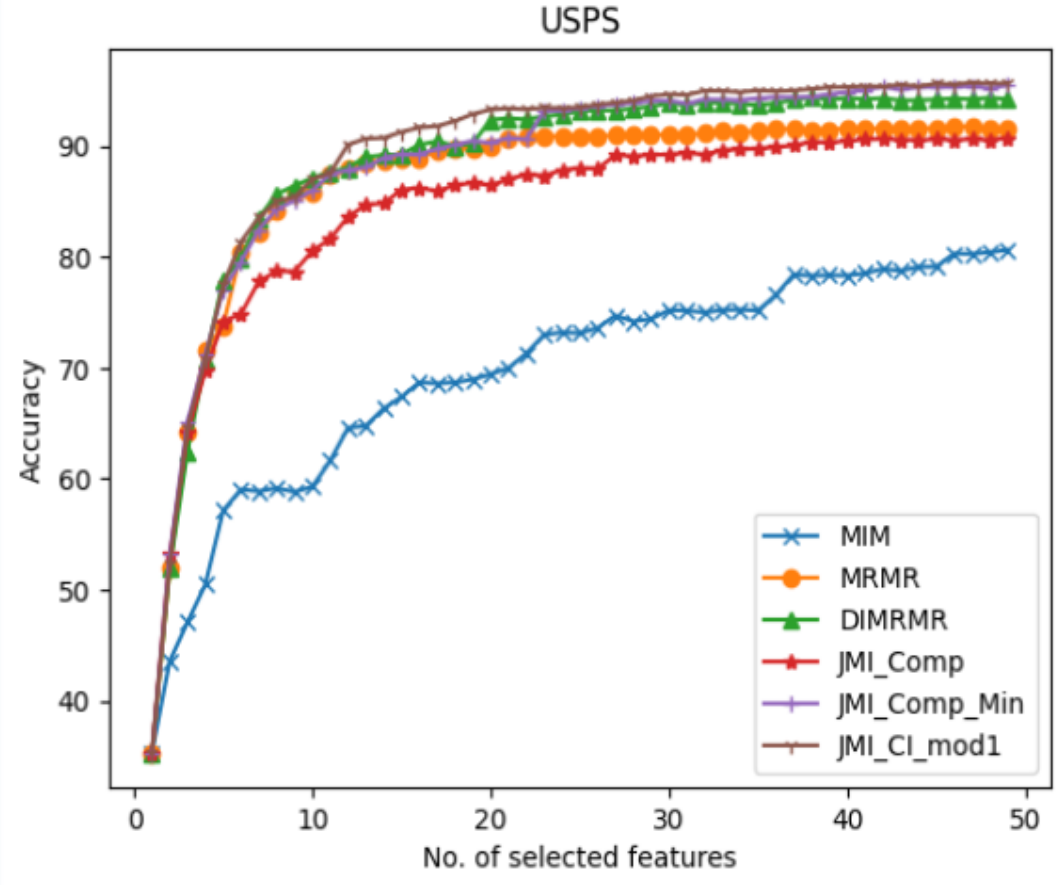

Figure 4.1: USPS accuracy curve

For USPS dataset, our method JMI_{Comp_CI} performs better than DIMRMR after 11 features.

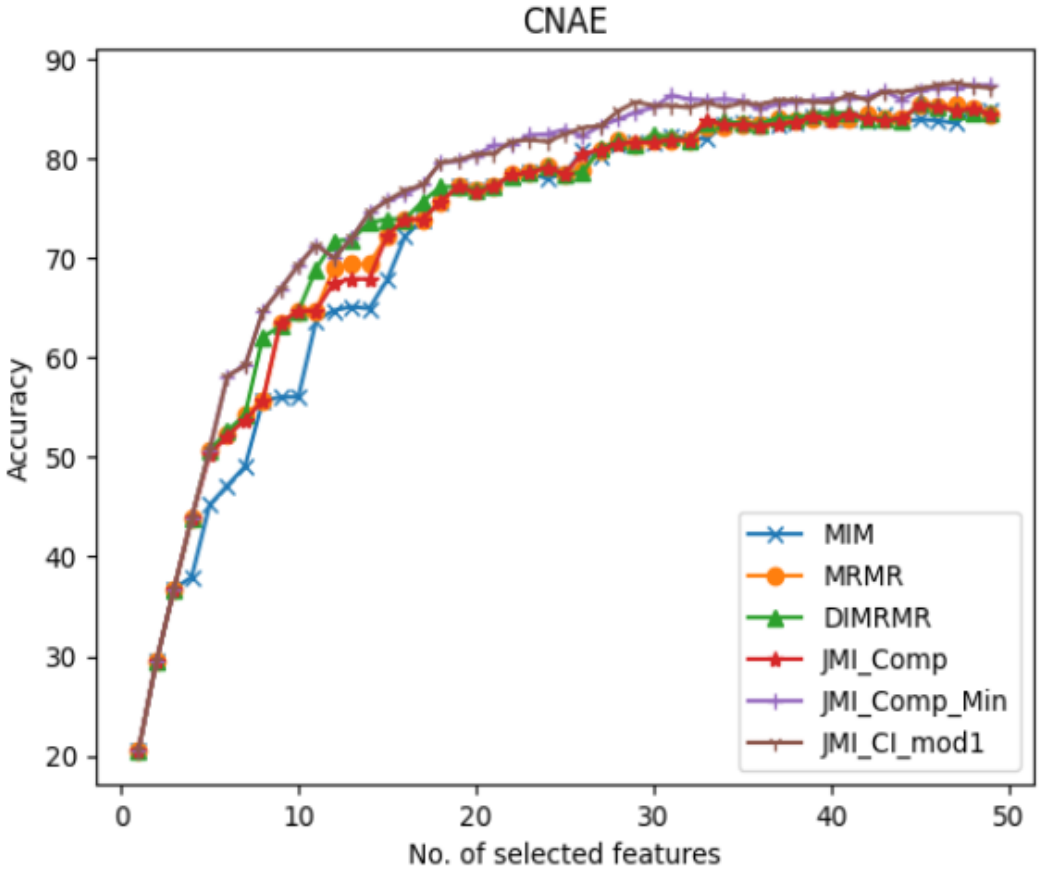

Figure 4.2: CNAE-9 accuracy curve

For CNAE dataset, our method JMI_{Comp_CI} and JMI_{Comp_min} both perform better than DIMRMR after 15 features.

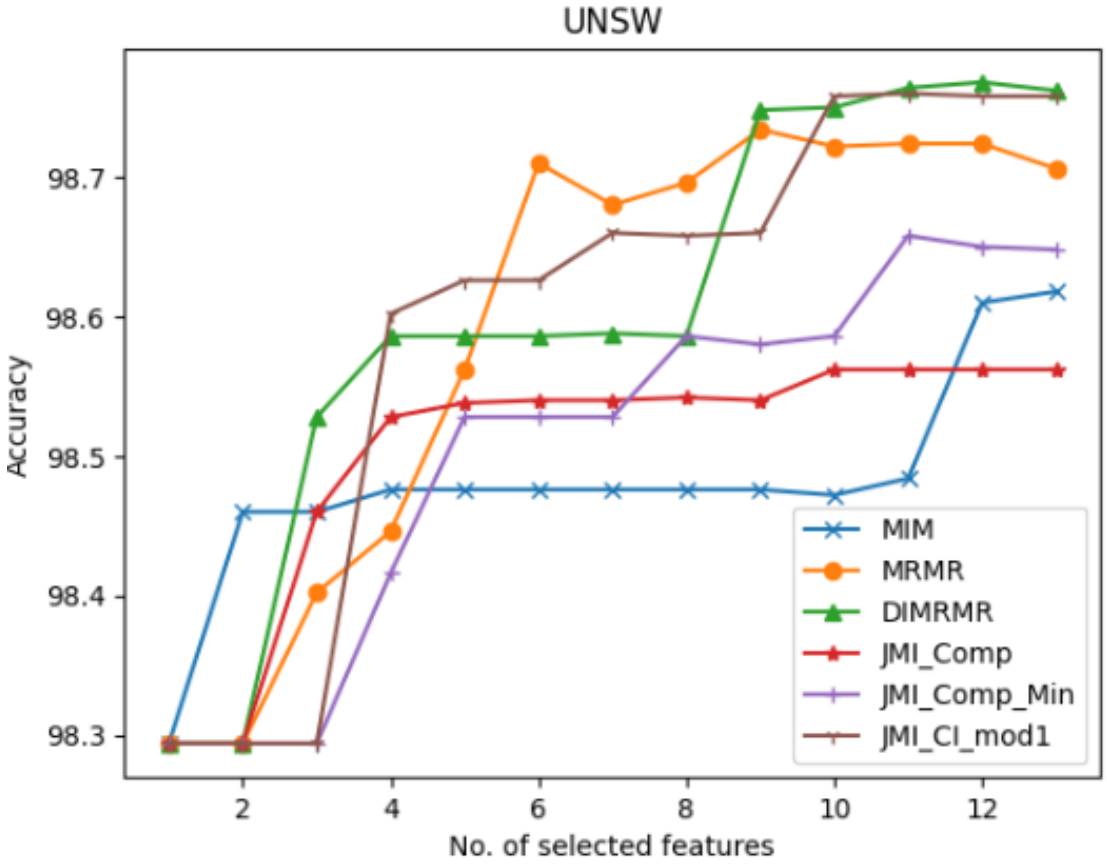

Figure 4.3: UNSW NB-15 accuracy curve

In case of UNSW NB-15 dataset, we can observe some anomalies in the graph. It is interesting that MRMR performed better than DIMRMR in cases of 6, 7 and 8 features. Our method performed better than DIMRMR in 4 to 8 features.

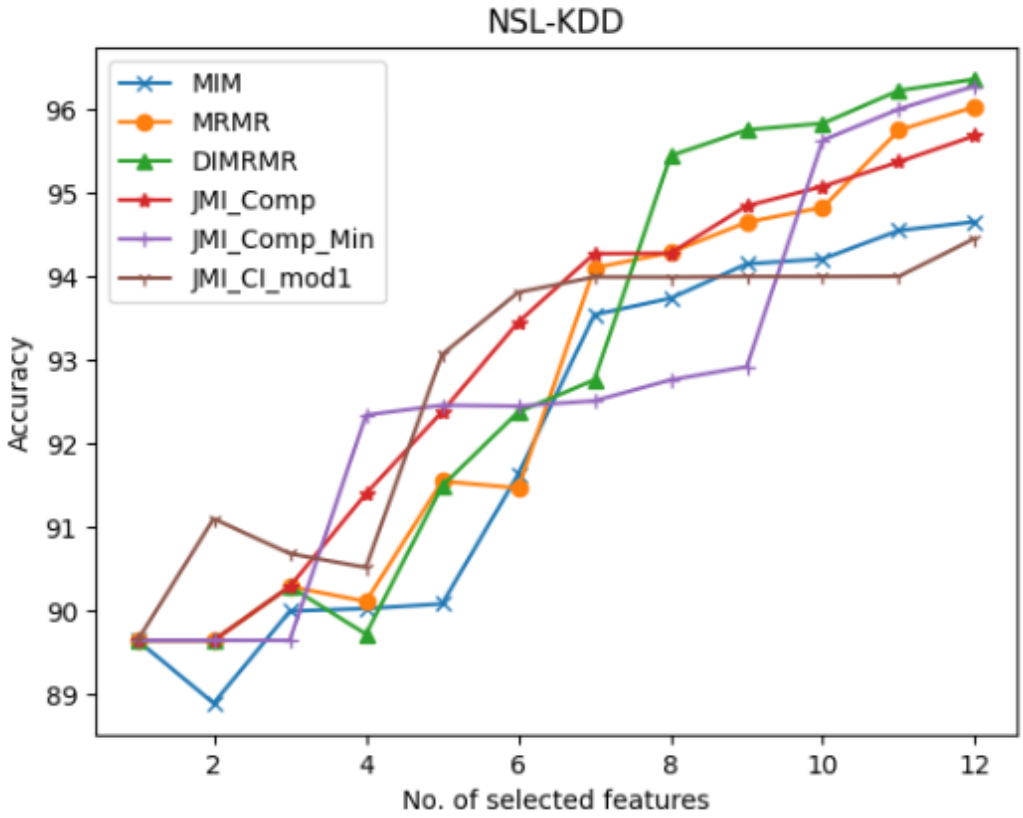

Figure 4.4: NSL-KDD accuracy curve

In case of NSL-KDD dataset, similar anomalies are spotted and we can see that our method performs better than DIMRMR in between 2 to 7 features.

The trend of the accuracy curve changes significantly when the methods are applied for network intrusion datasets. We can see anomalies in both UNSW and NSL-KDD datasets. The generic datasets' accuracy curves showed a gradually upward moving trend. But the network datasets present accuracy curves with although showing an upward trend but having irregularities in the curve. Such irregularities can be pointed out as straight line in the curve for a number of added features suggesting adding the features did little to zero contribution to the accuracy. Sometimes we can observe that the curve goes downwards after adding a feature suggesting adding a feature decreased the accuracy. These phenomenons are observed in cases of the literature methods such as MRMR, DIMRMR and also for our methods.

The anomalies spotted are happening most likely because the network datasets features have some complex relationship among them, and that is why we see different behavior than the generic datasets. This observation is supported by the fact that we can see flat lines in the accuracy curve although several more features are being added and in some cases accuracy drop after adding a feature, such as the case in UNSW dataset for MRMR at 7th feature, the case in NSL KDD for our JMI_{Comp_CI} method in 3rd and 4th features.

The following table signifies the overall performance of our methods by showing the mean accuracy with standard deviation. A larger mean value means this method performs well in most of the cases. The anomaly in case of the network datasets can also be seen by the higher standard deviation values.

	UNSW-NB15	**NSL-KDD**	**CNAE-9**	**USPS**
MIM	98.48 ± 0.07	92.10 ± 2.14	71.83 ± 16.27	69.64 ± 10.23
MRMR	98.59 ± 0.16	92.70 ± 2.36	73.14 ± 15.28	86.75 ± 10.57
DIMRMR	***98.60** ± 0.16	92.97 ± 2.68	73.70 ± 15.12	88.46 ± 11.27
JMI_Comp	98.50 ± 0.09	**93.03** ± 2.17	73.03 ± 15.33	84.39 ± 10.26
JMI_Comp_Min	98.51 ± 0.13	92.70 ± 2.25	**75.98** ± 15.60	88.60 ± 11.24
JMI_CI	***98.60** ± 0.17	92.78 ± 1.67	**75.98** ± 15.60	**89.42** ± 11.47

Table 4.2: Comparison of Average Accuracy(%) over the accuracy curves with standard deviation. Bold values indicate the highest accuracy for each dataset. Tie is indicated by *

Chapter 5

Conclusion and Future work

Conclusion

To sum up, objective of this thesis can be stated as two parts: to find a better MI based feature selection algorithm in general and apply the algorithm for NIDS datasets.

Our algorithm outperforms the latest state of the art MI based algorithm, DIMRMR, in the generic datasets that are used in related papers. The task of application for NIDS systems is also done but the behavior observation and results are not similar to the generic datasets.

Feature selection in NIDS goes beyond simple data reduction; it creates a watchful guardian from the chaos of raw data. Through the process of carefully selecting informative aspects and eliminating redundant information, it provides NIDS with accurate threat detection, rapid reaction, and proactive flexibility. Enhanced understanding of network behavior exposes sophisticated attack methods, informing strategic defense planning and ensuring the Network Intrusion Detection System (NIDS) remains effective against evolving cyber threats.

Future Work

The specifications of network related datasets require more study on how each feature is related to the others. This study would also invoice domain-specific knowledge. Above all, more precise and accurate algorithm of feature selection for

NIDS datasets can be designed.

Understanding how NIDS dataset features relate to each other is crucial for developing effective intrusion detection algorithms. Correlation analysis can play a significant role in identifying which features tend to vary together during normal versus anomalous traffic patterns.

Feature importance assessment is another critical aspect of inter-feature relationship study. By analyzing which features contribute most significantly to detecting various types of intrusions, NIDS systems can focus on the most relevant data points. Temporal relationships also warrant investigation, as understanding how feature values change over time can reveal patterns indicative of suspicious activity. Furthermore, contextual dependencies are essential to consider, as the importance of certain features may change based on factors like network topology or time of day.

Further root cause analysis can be done on the irregular behaviour of the accuracy curve when feature selection methods are applied in NIDS datasets.

Bibliography

[1] K. Yin, A. Xie, J. Zhai, and J. Zhu, "Dynamic interaction-based feature selection algorithm for maximal relevance minimal redundancy," *Applied Intelligence*, vol. 53, no. 8, pp. 8910–8926, 2023.

[2] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[3] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.

[4] H. Zhou, X. Wang, and R. Zhu, "Feature selection based on mutual information with correlation coefficient," *Applied Intelligence*, pp. 1–18, 2022.

[5] D. Lewis, "Feature selection and feature extraction for text categorization," *HLT '91: Proceedings of the workshop on Speech and Natural Language*, pp. 212–217, 10 2000.

[6] D. D. Lewis, "Feature selection and feature extraction for text categorization," in *Speech and Natural Language: Proceedings of a Workshop held at Harriman, New York, February 23-26, 1992*, 1992.

[7] F. Macedo, R. Valadas, E. Carrasquinha, M. R. Oliveira, and A. Pacheco, "Feature selection using decomposed mutual information maximization," *Neurocomputing*, vol. 513, pp. 215–232, 2022.

[8] M. H. Tarek, M. M. H. U. Mazumder, S. Sharmin, M. S. Islam, M. Shoyaib, and M. M. Alam, "Rhc: Cluster based feature reduction for network intrusion detections," in *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*. IEEE, 2022, pp. 378–384.

[9] T. Salman, D. Bhamare, A. Erbad, R. Jain, and M. Samaka, "Machine learning for anomaly detection and categorization in multi-cloud environments," in *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*. IEEE, 2017, pp. 97–103.

[10] L. Kurgan and K. J. Cios, "Caim discretization algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 145–153, 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:30857

[11] N. Dominique and Z. Ma, "Enhancing network intrusion detection system method (nids) using mutual information (rf-cife)," in *Security with Intelligent Computing and Big-data Services: Proceedings of the Second International Conference on Security with Intelligent Computing and Big Data Services (SICBS-2018) 2*. Springer International Publishing, 2020, pp. 329–342.