Chapter 5

MICC-UD: A Mutual Information and Correlation-based Feature Selection Method

5.1 Introduction

In feature selection, a subset of features are selected from an original set of features in machine learning or statistics [181]. Over the years, both supervised and unsupervised feature selection methods have been proposed. Supervised machine learning methods have garnered tremendous popularity and usage compared to unsupervised feature selection methods. Feature selection methods enhance the learning process by improving model performance enabling better visualization and understanding of the data, and better utilization of computational resources [128]. A good feature selection method requires the selected feature subset to be highly relevant to the target class. The features among themselves on the other hand, should have less redundancy among themselves. Feature relevance is important because it helps in distinguishing among the target classes. Otherwise, the feature does not play a contributing role in predicting the target class for a given test instance. Too many irrelevant features may also lead to the problem of overfitting. Unnecessary features make the learning model unnecessarily complex and difficult to understand. On the other hand, the candidate features in a selected subset should be irredundant of each other because redundant or dependent features do not contribute any additional knowledge to the learning process and as a result may bring down model performance.

Feature selection is an optional step before classification or prediction (in case of supervised learning). Although, an optional step, many studies in the literature have shown that the process of feature selection helps in attaining improved classification performance with reduced execution time. Moreover, in domains such as network security where detection of malicious attacks is of outmost importance in near real time, feature selection techniques may play a major role.

5.1.1 Feature

A feature, also called an attribute or variable, essentially describes the characteristics of a data point. For example, if one considers a person to be a data point, then height, weight, facial shape, hair color or eye color may be relevant features. All features taken collectively describe the data point for the task at hand. Features that are computed or derived from other features are called dervied features.
Particularly, in network security, the concept of features is not that simple. This is because from a single network packet, tens, hundreds or even thousands of features can be extracted. From these features, several other features may be derived in turn.

5.1.2 Related Work

In the literature, supervised feature subset selection methods are common. Optimal feature subset selection is the main focal point of these methods with an aim to achieve best possible classification performance [127, 182, 183]. Unsupervised feature selection methods are also on the rise [184–186]. However, such techniques are out of scope for this work. Irrespective of whether any knowledge is used or not, many filter, wrapper, embedded and ensemble feature subset selection methods are very popular in the literature.
Filter methods rely on statistical measures such as information gain, correlation, and mutual information to provide an ordered list of feature ranks [127, 187, 188]. These ranking schemes help highlight which features play important roles. Irrelevant features are filtered out and removed before performing the classification task since their presence degrades the quality of the feature set. Many filter methods are also used in conjunction with population-based heuristic search approaches to leverage the benefits of competitive ranking [189–192]. Mutual Information Feature Selection (MIFS) method is a very popular method which makes use of feature-

feature as well as feature-class mutual information for selecting a feature subset that maximizes the classification [193]. RELIEF is another popular technique where features are ranked according to a relevance criterion [194]; however, a threshold needs to be set for discarding a set of the irrelevant features. On the other hand, Bhatt et al. [195] and Cekik et al. [196] use fuzzy rough set and rough set theory respectively for feature subset selection. Using variable complementary measure Meyer and Bontempi [197] proposed Double Input Symmetrical Relevance (DISR), which is typically a filter feature selection method. The method tries to find intrinsic features from the data which has more information to identify the target class for a given instance. Song, Ni and Wang [198] utilized the concepts of graph theory in the proposed feature subset selection algorithm called Fast Clustering-based Feature Selection Algorithm. Firstly, features are clustered into different groups using Minimum Spanning Tree. Secondly, a measure called Symmetric Uncertainty measure [199] is used to eliminate the irrelevant and redundant features. Lastly, cluster-based methods are used to find the optimal subset of features from the original feature set. Another interesting method is the Conditional Mutual Information Maximization (CMIM) method [200], which tries to choose only those features that maximize the feature-class mutual information in an iterative manner provided the information of the selected features so far. Perhaps one of the pioneer filter methods which introduces the concept of feature-feature correlation is the Minimum Redundancy - Maximum Relevance method (mRMR) [201]. The main idea behind is to choose uncorrelated but highly informative features using the correlation measure. Sequential selection methods and Heuristic search methods [129, 130, 188, 202, 203] are two categories under Wrapper feature subset selection methods that are prevalent in the literature. Maldonado and Weber [204] propose a sequential backward selection wrapper method using Support Vector Machines (SVMs) to select features with fewer errors in a validation subset. Gang and Jin [205], use cosine similarity along with SVMs to select relevant and irredundant features. On the other hand, a hybrid feature selection method is proposed by Hsu et al. [206], where the filter methods help find the candidate features efficiently and then the wrappers are responsible for providing the classification results.

Ensemble feature selection methods such as [207] rely on base feature selection methods to provide individual lists of ranked features. These individual ranked features are then combined in some manner and the end result is final ranked list of features. Tsymbal et al. [208] considers diversity measures to quantify diversity in the ensemble feature selection techniques in conjunction with four search strate-

gies namely ensemble forward and backward sequential selection, hill-climbing and genetic search.

5.1.3 Limitations of the Existing Approaches

In the literature, a good number of mutual information and correlation based feature selection methods have been proposed. When considered individually, these methods are not free from limitations. Some of the common limitations are outlined below.

1. Univariate feature selection methods consider the correlation of a particular feature with the target class only [209]. The feature-feature correlation is not considered which is important because redundancy amongst the features should be less [210, 211] or in other words, features should be independent among themselves.

2. Correlation-based feature selection methods are helpful when the relationship between a pair of entities (features) is linear [212, 213]. However, the relationships among the real world entities may not always be linear. Mutual information based methods on the other hand, can also handle entities with non-linear and complex relationships [214–216].

3. Correlation-based methods are highly sensitive to outliers which can incorrectly impact the selection of features [217]. On the other hand, Mutual information based methods are less sensitive to outliers and hence even in the presence of noise/outliers such methods can identify valuable features[216].

4. Most Correlation-based methods consider the relationship between a given feature and the target variable only. How the features behave when taken in a combination is not taken into account, which is important because features among themselves may be co-related. Highly co-related features do not provide any new information.

From the above limitations it can be understood that depending only on mutual information or only on correlation may not be sufficient to make an informed decision on which features to select. To utilize the benefits of both, a feature selection method is proposed which combines both mutual information and correlation measure effectively to obtain a comprehensive subset of relevant and irredundant features. So, two points can be highlighted as below:

- Select relevant features by utilizing feature-class mutual information.

- Select features which are irredundant among themselves using feature-feature correlation.

5.1.4 Motivation

In recent times, the amount of available data has grown tremendously, in all domains including network security, bioinformatics, text categorization, and computer vision, to name a few in terms of number samples and dimensions. Although, data are generated in large amounts, not all of the data are of sound quality and ready for predictive data analysis. Machine learning methods, require relevant, easily understandable, meaningful, complete and recent data to provide significant insights to predictive modeling. To this end, feature selection, essentially a crucial pre-processing step. It helps a learning model simplify the learning process, and thereby gain meaningful and necessary knowledge for predictive tasks. Although, numerous feature selection techniques have been proposed, their ability to minimize false positives remains a major issue, specifically for the network security domain. Reducing irrelevant and redundant features from a feature set that characterizes a network instance leads to not only better accurate predictions, but also reduces computational time. All these reasons collectively, have motivated the development of a feature selection technique which focuses on selecting relevant and irredundant features, constituting an optimal feature subset, to achieve best predictive performance.

5.1.5 Contribution

The primary contribution is in the form of a mutual information and correlation based feature subset selection method called MICC-UD for identifying a subset of highly relevant and irredundant feature subset so as to obtain best possible classification accuracy. The optimality of the feature subset given by MICC-UD in terms of cardinality is substantiated using a total of 16 datasets. The proposed method has been found effective in achieving best possible classification accuracy using a number of prominent ensemble classifiers. Table 5.1 depicts the symbols and notations used to describe the proposed method.

Table 5.1: Symbol Table for the Proposed Method (MICC-UD)

Symbol	Symbol Meaning	Symbol	Symbol Meaning
D	Dataset	R	Random variable
d	Dimension of dataset D	S	Random variable
F	Original feature set of D	r	Marginal probability distribution of R
F'	Contains features whose mutual information with target class is greater than 0	s	Marginal probability distribution of S
		p(r,s)	Joint probability distribution of R and S
f_i	Feature number i	FFCorr	Correlation matrix
f_j	Feature number j	FCRel	Feature class relevance
M_p	Predictive model	AvgCorr	Average correlation
C	Target class	FFCClist	List containing correlation values for features
I	Mutual Information	max_corr	Maximum correlation

5.2 Problem Formulation

Let's assume a Dataset D, with feature set $F = \{f_1, f_2, f_3, ...f_d\}$, where d is the dimension of the dataset. The main aim is to choose an optimal feature subset, F' of relevant and irredundant features. Here, $F' \subseteq F$ and F', is obtained to give the best possible prediction for a predictive model M_p. Therefore, the goal is to determine a feature subset F' containing features with high feature-class mutual information, (f_i, C) i.e. high relevance and low feature-feature (f_i, f_j) correlation (irredundant features).

5.3 Background

The proposed method is designed by exploiting two popular yet powerful statistical measures namely Mutual Information and Correlation. The subsequent sections discusses briefly on the two measures and their usefulness in feature selection.

5.3.1 Mutual Information for Feature Selection

Let us assume that R and S are two random variables. The amount of information that R holds about S can be termed as *Mutual Information*[218]. Mathematically, this can be conveyed as in Equation 5.1.

$$I(R;S) = \sum_{r,s} p(r,s) \log \frac{p(r,s)}{p(r)p(s)} \tag{5.1}$$

As expressed, $p(r,s)$ in Equation 5.1 is the joint probability distribution function for the random variables R and S. On the other hand, $p(r)$ and $p(s)$ signify the marginal probability distributions for R and S. It is important to note that, the Mutual Information between R and S is said to be zero if they are statistically independent.

The idea of mutual information is originally related to entropy, which helps measure the amount of uncertainty that one can expect in a random variable[219][220]. Equation 5.2 highlights the relation between mutual information and entropy.

$$I(R;S) = H(R) - H(R|S) \tag{5.2}$$

Here, H(R) is the marginal entropy, which signifies distribution specific information with regards to random variable R. On the other hand, the conditional entropy which is expressed by $H(R|S)$ measures the uncertainty in R due to the knowledge of S. Intuitively, this is how $I(R;S)$ explains the connection between the random variables R and S in terms of entropy.

For feature selection, Mutual Information is significant because it helps to establish how relevantly a particular feature (or attribute) is related to the target class. In other words, it helps find how useful a feature is in predicting a target class. A feature, say f_i, which has a higher mutual information score with the target class compared to a feature say f_j, will be hence, more useful in predicting the target class as it will have more information regarding the target.

In information theory, mutual information measures how two quantities are related. I(R,S), the mutual information between random variables R and S measures the uncertainty in R due to the knowledge of S as expressed in Equation 5.1.

5.3.2 Correlation Co-efficient for Feature Selection

In feature selection, correlation which is a statistical measure is used to find how two features, say f_i and f_j, are related to each other. Two features can be positively,

negatively or not co-related at all, i.e., correlation value may be positive, negative or even zero. If strongly related, the presence of either one of them in the feature set is enough. On the other hand, presence of both features would result in a redundant feature set. Therefore, the aim is to choose a subset of features from the original feature set in such a way that the redundancy amongst these features is minimum.

Pearson's correlation coefficient (given by Equation 5.3) describes two entities say R and S, and the linear relationship between them. The correlation coefficient gives a value between -1 and +1. A value of -1 signifies strong negative correlation and a value of +1 signifies strong positive correlation relationship. Moreover, a value of 0 signifies no correlation. For MICC-UD, the correlation coefficient's absolute value is considered as the goal is to determine the strength of the relation only and not whether it is positive or negative.

$$Corr - Coeff = \frac{\sum (r_i - \bar{r})(s_i - \bar{s})}{\sqrt{\sum (r_i - \bar{s})^2 \sum (s_i - \bar{s})^2}} \tag{5.3}$$

5.4 MICC-UD: Proposed Method

MICC-UD comprises both mutual information and correlation measures to obtain a final ranked list of features. Figure 5.1 illustrates the proposed framework.

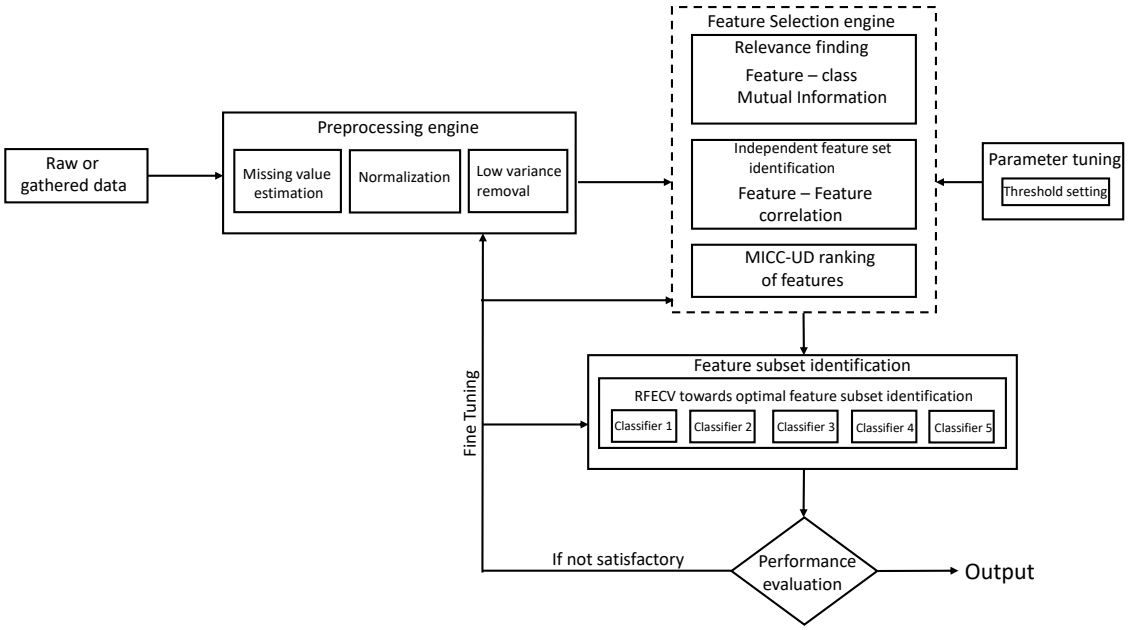

Figure 5.1: Proposed framework for MICC-UD

5.4.1 Preprocessing Engine

After data is gathered/generated, preprocessing tasks are carried out by the preprocessing engine. This engine comprises of three sub-modules. Missing values, if any in the data, are either estimated or the example is removed. Missing value estimation is performed by averaging the column values. Next, features with low variance are removed as they do not contribute much to the decision making process. The third step in this engine is the normalization step, for which min-max normalization is used.

5.4.2 Feature Selection Engine

Next in the framework is the feature selection engine comprising three sub-modules: Relevance finding sub-module, Irredundant feature set identification sub-module and the MICC-UD feature ranking sub-module. The purpose of the feature selection engine is to select the subset of features which are highly relevant and irredundant.

5.4.2.1 Relevance Finding

The first sub-module calculates feature-class mutual information using Equation 5.1. This helps find the highly relevant features for the prediction task. A feature say f_i is said to be highly relevant if it has high mutual information with the target class, say C. On the other hand, the feature f_i is regarded as not relevant if it has zero mutual information with the target class C. Such a feature is dropped from the final list of features as such a feature will play no role in the prediction task.

Definition 5.1. (Feature Relevance) The relevance of a feature f_i with respect to a target class C is defined in terms of mutual information between the feature and the class. Higher the mutual information score for a feature f_i with C higher is its relevance.

5.4.2.2 Irredundant Feature Identification

The second sub-module is responsible for identifying the irredundant features by calculating the feature-feature correlation. Pearson correlation is used to find the dependency between two features using Equation 5.3. Pearson's correlation gives a value between -1 and +1. However, the correlation score's absolute value is

considered as the point of interest is the strength of the relationship between the two features and not the magnitude (positive or negative) of the strength.

Definition 5.2. (Feature Redundancy) For a feature f_i, its redundancy is defined in terms of average correlation with all other features in F i.e. f_1 to f_d, where $i \neq d$. Lower the average correlation score for a feature f_i, lower is its redundancy.

Definition 5.3. (Average Correlation of a feature) The average correlation of a feature f_i, is defined in terms of the summation of correlation values between feature f_i and feature f_j (where $f_j \in F, i \neq j$ and $j \leq d$) divided by d, where d is the total number of features in F.

5.4.2.3 MICC-UD Ranking of Features

The third sub-module is the heart of the framework and computes the rank of each feature according to the Equation 5.4. The feature selection engine works in conjunction with a parameter tuning module which is responsible for providing as input a threshold value. Only those features whose score as given by MICC-UD is greater than a particular threshold value are chosen for prediction.

$$MICC\text{-}UD(f_i) = \frac{Relevance\_score(f_i, C)}{max_{i \neq j}(|avg\_Corr|(f_i) - Corr(f_i, f_j))} \tag{5.4}$$

The relevance score and average correlation (avg_Corr) mentioned in Equation 5.4 is calculated as shown below in Equation 5.5 and Equation 5.6.

$$Relevance\_score(f_i, C) = MutualInformation(f_i, C) \tag{5.5}$$

$$avg\_Corr(f_i) = \frac{\sum_{\substack{j=1 \\ i \neq j}}^{d} (Corr(f_i, f_j))}{d} \tag{5.6}$$

Definition 5.4. (MICC-UD score) The MICC-UD score is defined as the mutual information correlation coefficient for a given feature f_i, which is estimated in terms of both relevance and (mutual information) and irredundance (average correlation), where f_i should have high relevance with the target class C and at the same time should be less redundant on the other selected features from F.

Specifics of MICC-UD: To assess the strength of the relationship between two features correlation is used, specifically Pearson's correlation coefficient. Higher strength indicates that the features are highly correlated. The proposed method uses Pearson's correlation coefficient because of three main reasons: i) It measures

the relationship between two continuous variables (raw values of the variables), unlike other correlation measures such as Spearman Rank Correlation which takes into consideration the ranks of the data or Kendall's Tau correlation measure which considers the ordinal association between two variables [221], *ii*) It is a widely accepted standard measure and hence is not influenced by the scales of the continuous valued features [222], *iii*) Pearson's correlation coefficient is simple and fast in terms of computational complexity ($\mathcal{O}(n)$) compared to Spearman's coefficient ($\mathcal{O}(n \log n)$) [223]. Additionally, as already mentioned that correlation measures (in this case the Pearson's correlation measure) are sensitive to outliers. This drawback is overcome with a two fold solution. First, mutual information (as it is insensitive to outliers [216]) is introduced to the proposed score. Second, to negate out the outlier effects of a feature say f_j when calculating the correlation with feature f_i, the average correlation of f_i is introduced and subtraction of the two entities is performed as shown in Equation 5.4. This ensures that the outlier values of f_j will not influence the values of feature f_i.

Initially, a feature f_i's relevance with the target class is calculated using feature-class mutual information. All features which have zero relevance are removed. So, the candidate feature set now includes only those features which have relevance to the target class. For each feature, say f_i currently present in the candidate set, the pairwise correlation with all the other features f_j in the candidate set is determined. The average correlation of each feature f_i is next. The list FFCClist (initially empty) consists of values which signify the difference between the average correlation of feature f_i and the pairwise correlations of f_i with f_j. From this list, the maximum value, max_corr is chosen for f_i. The final score for each feature is the relevance score divided by the max_corr value of that feature. Finally, a ranked list is obtained which consists of relevant features and their MICC-UD scores.

5.4.3 Optimal Feature Subset Identification using Recursive Feature Elimination

After obtaining the ranked list of features, next the optimal feature subset needs to be identified. Optimal subset of features mean adding features to this subset does not increase the classifier performance and at the same time, removing any feature from the subset deteriorates the classifier performance. Here, the optimal feature

subset is obtained by recursively eliminating the features from the ranked list. The recursive feature elimination step works primarily with five predictors for making predictions. After the final predictions are obtained, performance of the predictors are evaluated and if not satisfactory, any of the two engines namely preprocessing engine or feature selection engine or the feature subsets identified may have to be fine tuned.

5.4.4 Proposed Algorithm

MICC-UD relies on computing feature-class mutual information using function *MutualInformation*, feature-feature correlation using function *Correlation* and the proposed score which calculates the final rank for each feature f_i. The input is a dataset D with feature set $F = \{f_1, f_2, \cdots, f_d\}$. The output is a ranked list of features containing each feature's MICC-UD score. The MICC-UD method is described in Algorithm 2.

For a feature $f_i \in F$, the *MutualInformation* function computes the relevance with the target class C in terms of Mutual Information as given in Equation 5.1. This module is also responsible in removing features which have zero relevance with the target class. After removing such features, a set $F^{'}$ is obtained. For two features $f_i, f_j \in F$, the *Correlation* function computes the correlation between the features using Equation 5.3. For each feature f_i, the MICC-UD score according to the formula given in Equation 5.4 is calculated.

Proposition 5.1. Features selected by MICC-UD are relevant.

Proof. Let F' be a subset of features selected by MICC-UD, where $F' \subseteq F$, i.e., the original set of features. Let $f_i \in F'$, be a selected feature, which is not relevant. MICC-UD selects a feature iff it has high relevance with a given class. In other words, $f_i \in F'$ only when the relevance, i.e., mutual information score for f_i is significantly high. Therefore, the assumption contradicts. Hence, the proof. ■

Proposition 5.2. Two features f_i and f_j are included in F' if they are less redundant with each other.

Proof. Let $(f_i, f_j) \in F'$ be a pair of features included in the selected feature subset F', and let us assume (f_i, f_j) are redundant. The proposed method selects a feature

Algorithm 2: MICC-UD

Input : Dataset D, with dimension d, and feature set $F = \{f_1, f_2, \cdots, f_d\}$

Output: Ranked list of features

Steps:

for *i=1 to d,* **do**
    Calculate `MutualInformation` (f_i, C)
end
Select feature f_i with `MutualInformation` $(f_i, C) > 0$
$F' = F' \cup \{f_i\}$
for *i=1 to d* **do**
    for *j=1 to d* **do**
        FFCorr = Calculate `Correlation` (f_i, f_j)
    end
end
for *each feature* $f_i \in F'$, **do**
    FCRel (f_i) = Calculate `MutualInformation` (f_i, C)
    AvgCorr = average correlation of feature f_i with other features
    FFCClist = |AvgCorr − $FFCorr$|
    Select for feature f_i, max_corr, the maximum correlation from list FFCClist
    MICC-UD(f_i) = $\frac{\text{FCRel}(f_i)}{max\_corr(f_i)}$
    Ranked_list = MICC-UD(f_i)
end
Return Ranked_list

f_i only if it has lower correlation with the other selected features in F'. From Equation 5.4, it is clear that the score value of MICC-UD is inversely proportional to the correlation score. For a given pair of features (f_i, f_j) and for a given relevance score of f_i, if the correlation is high, then the overall score will be less and consequently it will not be selected. It contradicts the assumption. Hence, the proof. ■

5.5 Complexity Analysis

The governing factor in calculating the complexity of MICC-UD is in the construction of the correlation matrix incorporating the correlation values of each feature

f_i with f_j. Larger the dimension of the dataset D, larger the correlation matrix. Thus, complexity of the algorithm largely depends on the input dataset. If the input dataset is of dimension d, the complexity of MICC-UD is given by $\mathcal{O}(n^2)$. MICC-UD relies on constructing a correlation matrix for finding irredundant features from the original feature set. Even though, to build the correlation matrix only those features are considered which have mutual information with target class greater than 0, the size of the matrix is still governed largely by the dimensions of the input data. This is why, the complexity of the proposed method is in the factor of (n^2). However, this complexity could be reduced further if the proposed method could use parallel implementations to build the correlation matrix among the features. Such implementations are to be incorporated in future work to enhance MICC-UD.

5.6 Experimental Results

In this section, the datasets used to evaluate the proposed work are discussed and the experimental results are presented. Table 5.2 gives description of the datasets used. Special importance is the Ransomware (multiclass) dataset, where there are a total of 11 ransomware families. To handle this special case, the dataset is divided into 11 different datasets each containing the normal class (0) and an attack variant. So, from a single multiclass dataset, 11 different datasets of varied number of instances are obtained. In this special case hence, for detailed analysis the results are presented in class-specific manner as well.

Table 5.2: Datasets Used

Sl No.	Dataset	No. of features	No. of instances	No. of classes
1	Android Dataset 1 [142]	242	2140	2
2	Android Dataset 2 [142]	348	1110	2
3	Kitsune Network Attack [145]	116	7,64,137	2
4	Phishing [144]	31	11,055	2
5	XSSD	14	6695	2
6	Ransomware (multiclass) [143]	30, 967	1524	11

5.6.1 Results and Analysis

For all datasets, after finding a ranked list of features according to the scores given by MICC-UD, recursive feature elimination is performed in association with five conventional well-known ensemble classifiers, namely Adaboost [175], Gradient Boosting [224], Extreme Gradient Boosting [225], Random Forest [226] and Extra Trees [227]. For the experiments ensemble classifiers are considered because such classifiers are known to provide stable and reliable performances compared to individual learners [120][122]. Ensemble learning relies on the idea that decision given by a group of experts is always better than the decision given by an individual. Errors or misclassifications made by any one of the base learner in the ensemble may be canceled out by other learners participating in the ensemble. This is how a stable and reliable outcome is reached.

All classifier performances are recorded in a 10-fold cross-validation setting. Eliminating features recursively helps find an optimal subset of features for which a stable classification performance is achieved. Here, optimal subset of features means a feature set from which adding or removing features does not show any improvement in the classification performance.

Figure 5.2 illustrates the result analysis of the proposed method with respect to three measures namely, Accuracy, Precision and Recall. From the results, it is seen that of all the five classifiers used, Extra Trees performs best for four out of the six datasets (namely, Android Dataset 1, Phishing, XSSD and Ransomware Multiclass Dataset). However, for Android Dataset 2 and Kitsune Network Attack Dataset, Random Forest gives the best classification performance. With regards to the Ransomware Multiclass Dataset, the results presented in Figure 5.2 specify the average accuracy, precision, recall for the whole dataset. Figure 5.3 presents the class-specific results of the Ransomware Multiclass Dataset. From the results it can be seen that, Extreme Gradient Boosting classifier gives better performance than the rest for four out of the eleven ransomware variants (namely variants Kollah, Kovter, Locker and TeslaCrypt). Random Forest classifier on the other hand, shows better performance for three ransomware variants (namely, variants CryptoWall, Pgpcoder and Trojan-Ransom).

Table 5.3 shows the top 10 ranked features for the XSSD dataset considered as given by the proposed feature selection method. The columns *Feature_ name* and *index* gives the name of the feature and the index number in the dataset.

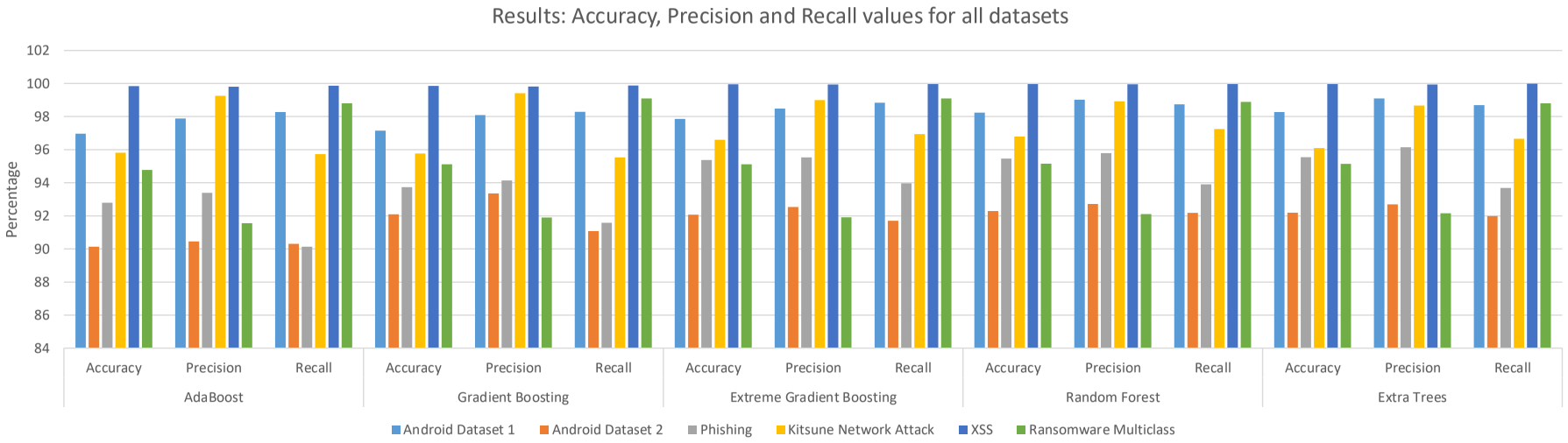

Figure 5.2: MICC-UD Results for all Datasets

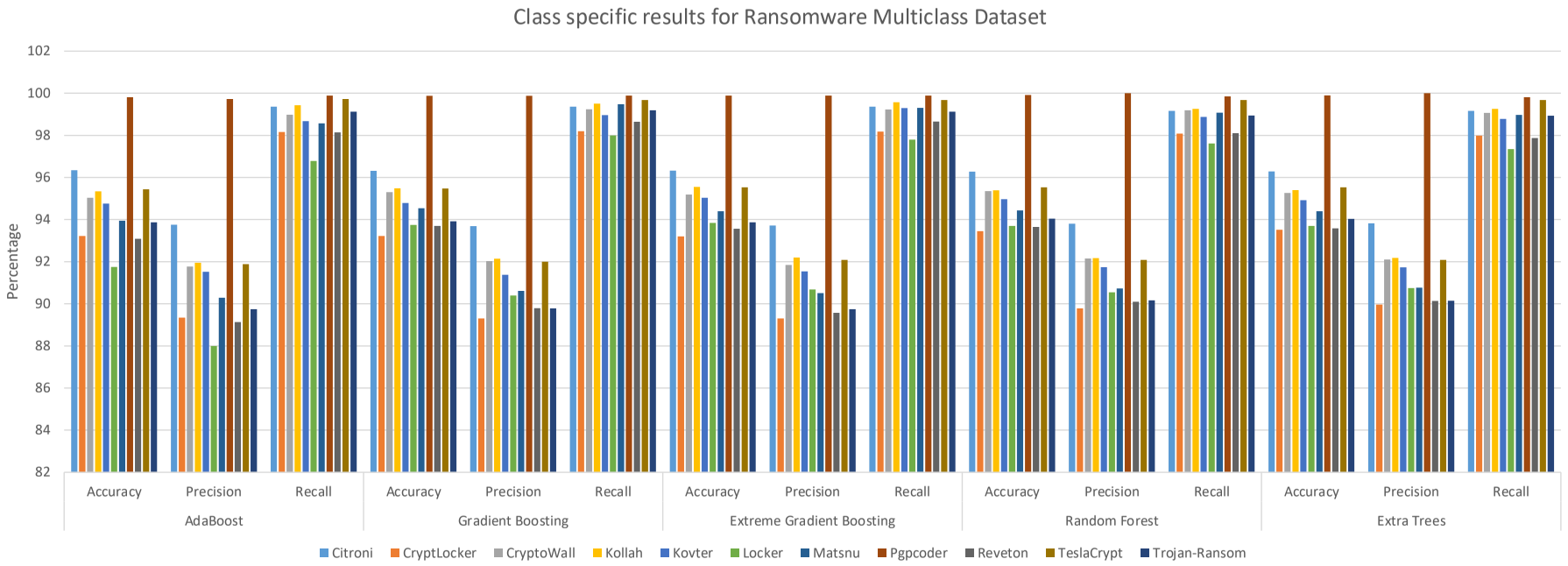

Figure 5.3: Class-specific Results for Ransomware Multiclass Dataset

Table 5.3: Top 10 Ranked Features of the XSSD Dataset

Feature_name	index
Keywords	10
No. Of Lines	1
No. Of words	6
NumberOfChars	0
No. of Unicode Symb	8
Avg Char per Line	2
No. Of methods called	7
No. Of Comment Lines	4
Avg Comment per Line	5
No. Of HEX symb	9

5.6.2 Comparison with Existing Works

In Table 5.4 and Table 5.5, a comparative analysis of MICC-UD is presented with other well-known feature selection methods such as MIFS [193], CMIM [200], and mRMR [228][201]. For the comparative analysis, we use F1-score instead of Accuracy as a measure. This is because F1-score is a better measure than Accuracy when it comes to unbalanced datasets.
In both the tables when comparing the methods, the average F1-score and the Number of Selected Features (NSF) using Recursive Feature Elimination is indicated. For example, in Table 5.4, it can be seen that in case of Android Dataset 1, for MICC-UD and AdaBoost classifier an average F1-score of 98.06% is obtained and the number of features selected to obtain the specified F1-score is 32. Here it is to be noted that for the Ransomware Multiclass dataset in Table 5.4, the F1-score comparisons of all the methods is reported on average across all the 11 classes. Here, in the table it can also be seen that for all the six datasets MICC-UD gives better performance in terms of F1-score measure for almost all the classifiers. However, in case of Android Dataset 2, XSSD and the Ransomware Multiclass datasets, when considering the number of selected features criteria it can be seen that MIFS outperforms MICC-UD as it selects less number of features for reporting the corresponding F1-scores. Therefore, it can be said that there is a trade off between both the criteria as the method which gives better performance in terms of F1-score may not always give the least number of features.

Table 5.5 reports the class-specific comparison of MICC-UD with three other state of the art feature selection methods. It can be seen from the results that for

Table 5.4: Comparison of F1-scores for all Datasets

Comparison of F1-scores for all Datasets											
		Adaboost		Gradient Boosting		Extreme Gradient Boosting		Random Forest		Extra Trees	
		F1-score	NFS	F1-score	NFS	F1-score	NFS	F1-score	NFS	F1-score	NFS
Android Dataset 1	CMIM	97.92	**30**	98.12	11	98.48	27	98.79	19	98.79	27
	MIFS	97.03	57	97.29	6	97.36	70	97.38	70	97.41	71
	MRMR	97.93	29	**98.26**	15	98.4	27	98.61	26	98.65	29
	MICC-UD	**98.06**	32	98.17	**4**	**98.63**	**25**	**98.87**	**14**	**98.89**	**12**
Android Dataset 2	CMIM	89.89	**4**	91.82	14	91.62	9	92.07	16	91.77	14
	MIFS	**91.07**	6	91.07	**5**	90.89	**5**	91.11	**6**	91.17	**5**
	MRMR	89.75	13	91.49	20	91.44	18	91.8	26	91.58	27
	MICC-UD	90.13	9	**92.02**	16	**91.98**	13	**92.28**	16	**92.17**	25
Phishing	CMIM	91.51	21	92.83	7	94.72	12	94.72	12	94.89	12
	MIFS	91.42	12	92.59	13	94.28	13	94.3	13	94.38	13
	MRMR	91.33	6	92.7	8	94.63	12	94.67	13	94.76	12
	MICC-UD	**91.72**	**3**	**92.83**	**6**	**94.73**	**9**	**94.82**	**9**	**94.89**	**10**
Kitsune Network Attack	CMIM	96.86	68	97.26	68	97.94	**68**	97.94	71	97.47	85
	MIFS	97.44	68	97.64	68	97.87	71	97.58	71	96.46	64
	MRMR	**97.53**	68	**97.72**	**68**	97.93	68	97.64	**68**	96.48	64
	MICC-UD	97.45	56	97.42	81	**97.95**	81	**98.08**	81	**97.66**	**9**
XSSD	CMIM	99.58	12	99.66	10	99.84	7	99.81	7	99.83	7
	MIFS	99.12	**4**	99.14	**4**	99.36	**4**	99.11	**4**	99.11	**4**
	MRMR	99.63	15	99.67	14	99.78	11	99.78	11	99.78	11
	MICC-UD	**99.84**	7	**99.85**	5	**99.95**	5	**99.96**	5	**99.97**	7
Ransomware Multiclass	CMIM	94.25	25	94.48	23	94.47	23	94.63	26	94.63	25
	MIFS	94.45	**12**	94.37	**11**	94.11	**11**	94.48	**12**	94.5	**12**
	MRMR	93.85	25	94.46	23	94.61	22	94.69	21	94.64	21
	MICC-UD	**94.98**	18	**95.33**	17	**95.33**	18	**95.33**	17	**95.31**	18

ransomware variants *Kovter* to *Trojan-Ransom* MICC-UD gives on par or better performance for most of the predictive models considering the F1-score measure. However, for variants *Citroni* to *Kollah*, with respect to F1-score criteria, CMIM method outperforms MICC-UD. Nonetheless, MICC-UD gives better performance than CMIM when considering the number of selected features criteria.

5.7 Discussion

MICC-UD helps in the identification of a feature subset comprising of relevant and irredundant features. The main idea is to choose features which have high

relevance with the target class but low correlation values with other features. To demonstrate the effectiveness of the proposed method, it is tested with 16 datasets and the results are reported in terms of accuracy, precision and recall. The results show that tree-based learners namely Extreme Gradient Boosting, Extra Trees and Random Forest achieves better prediction performance for the 16 datasets considered. However, when it comes to choosing the optimal number of features and the best predictive performance in terms of F1-score, there are trade offs. Depending on the situation, one needs to choose the significance between the predicted performance and the number of features selected. Additionally, a detailed comparative analysis is conducted with three benchmark feature subset selection methods and it is seen that MICC-UD performs better for most of the cases. Two limitations of the proposed method however are: i) it depends on a user-threshold as input to select the k-top features, ii) parallel implementations are not utilized to build the correlation matrix which can be very time consuming in case of large datasets. As future work, MICC-UD would have to be implemented in such a way that it is totally independent of the user-input. Also, more time efficient implementations will have to be introduced in building the correlation matrix.

The next chapter introduces an incremental feature selection method named INFS-MICC which is based on the foundations of MICC-UD. INFS-MICC is capable of differentiating between HTTP Flooding attack and normal HTTP traffic.

Table 5.5: Class-specific Comparison of F1-scores for Ransomware Multiclass Dataset

	Class-specific Comparison of F1-scores for Ransomware Multiclass Dataset										
		AdaBoost		Gradient Boosting		Extreme Gradient Boosting		Random Forest		Extra Trees	
	Feature Selection Methods	F1-score	NFS	F1-score	NFS	F1-score	NFS	F1-score	NFS	F1-score	NFS
Citroni	CMIM	96.46	13	**96.53**	16	**96.51**	14	**96.47**	9	**96.48**	9
	MIFS	96.08	**5**	96.44	5	96.17	6	96.24	7	96.38	7
	MRMR	96.28	11	96.3	12	96.34	9	96.26	8	96.27	7
	MICC-UD	**96.47**	10	96.45	**4**	96.44	**5**	96.38	**5**	96.39	**5**
CryptLocker	CMIM	94.95	17	**95.38**	11	95.5	12	**95.52**	11	**95.4**	11
	MIFS	**95.04**	16	95.07	**7**	95.14	**6**	95.1	8	95.21	7
	MRMR	95.03	19	95.09	18	**95.27**	18	95.19	18	95.04	18
	MICC-UD	93.49	**16**	93.49	9	93.48	7	93.69	**7**	93.74	**6**
CryptoWall	CMIM	**95.78**	38	**96.26**	34	**96.22**	37	**96.15**	37	**96.11**	37
	MIFS	94.76	14	93.79	11	93.57	11	94.36	15	94.36	15
	MRMR	95.22	46	95.76	26	95.78	15	95.67	16	95.59	15
	MICC-UD	95.22	**7**	95.48	**11**	95.38	**11**	95.52	**12**	95.44	**12**
Kollah	CMIM	**97.07**	23	**97.02**	22	**96.98**	21	**96.88**	21	**96.86**	21
	MIFS	96.45	16	96.31	**15**	96.14	**10**	96.48	17	96.46	16
	MRMR	96.49	**13**	96.58	18	96.54	18	96.3	**14**	96.26	**10**
	MICC-UD	95.53	28	95.67	28	95.73	28	95.57	27	95.57	28
Kovter	CMIM	91.77	25	91.68	13	91.62	25	92.03	25	92.05	25
	MIFS	94.2	10	94.24	10	93.86	**6**	94.15	**10**	94.16	**10**
	MRMR	91.45	34	92.24	36	92.39	36	92.52	36	92.53	37
	MICC-UD	**94.93**	**3**	**95**	**5**	**95.24**	12	**95.15**	12	**95.09**	12
Locker	CMIM	**92.2**	16	92.95	30	92.82	**11**	93.21	40	93.11	44
	MIFS	93.2	**13**	93.37	**7**	92.76	15	93.37	**8**	93.4	**8**
	MRMR	89.72	21	91.31	24	91.45	29	91.7	28	91.32	27
	MICC-UD	92.01	18	**93.98**	17	**94.05**	18	**93.88**	17	**93.86**	18
Matsnu	CMIM	91.26	26	91.58	22	91.74	23	91.83	23	91.92	22
	MIFS	**94.59**	12	94.4	12	94.14	18	94.41	18	94.42	22
	MRMR	91.98	41	92.39	28	92.75	26	92.89	26	92.96	26
	MICC-UD	94.19	25	**94.82**	26	**94.68**	26	**94.68**	26	**94.65**	26
Pgpcoder	CMIM	99.76	10	99.8	12	99.82	13	99.76	10	99.76	7
	MIFS	99.6	6	99.6	14	99.46	8	99.61	5	99.61	6
	MRMR	99.58	10	99.6	**10**	99.62	10	99.59	10	99.58	10
	MICC-UD	**99.81**	**5**	**99.88**	12	**99.89**	**6**	**99.92**	**5**	**99.9**	**6**
Reveton	CMIM	91.4	31	91.54	25	91.36	22	91.75	28	91.8	22
	MIFS	90.6	**15**	90.67	**11**	90.36	**10**	90.76	**11**	90.76	**12**
	MRMR	91.11	29	93.26	25	93.68	28	**93.93**	23	**93.97**	24
	MICC-UD	**93.37**	29	**93.97**	20	**93.87**	20	93.87	20	93.77	20
TeslaCrypt	CMIM	93.78	51	93.95	37	93.98	50	94.48	50	94.53	50
	MIFS	91.66	**19**	91.62	**18**	91.49	**18**	91.61	**19**	91.68	**19**
	MRMR	93.87	25	93.94	26	94.05	27	94.33	27	94.38	27
	MICC-UD	**95.64**	25	**95.67**	27	**95.72**	27	**95.72**	27	**95.72**	27
Trojan-Ransom	CMIM	92.35	28	92.63	27	92.68	27	92.89	28	92.92	28
	MIFS	92.83	**10**	92.63	**10**	92.19	**11**	93.29	**10**	93.07	**10**
	MRMR	91.62	28	92.6	25	92.91	25	93.21	25	93.22	25
	MICC-UD	**94.18**	34	**94.23**	32	**94.18**	34	**94.31**	32	**94.3**	34