A Machine Learning Approach to False Alarm Detection for Critical Arrhythmia Alarms

Xing Wang*, Yifeng Gao*, Jessica Lin*, Huzefa Rangwala*, Ranjeev Mittu†

*Department of Computer Science, George Mason University, Virginia, USA
†Naval Research Laboratory
{xwang24, ygao12, jessica}@gm.edu, {rangwala}@cs.gmu.edu, {ranjeev.mittu}@nrl.navy.mil

Abstract—High false alarm rates in Intensive Care Unit (ICU) is a common problem that leads to alarm desensitization—a phenomenon called alarm fatigue. Alarm fatigue can cause longer response time or missing of important alarms. In this work, we propose a methodology to identify false alarms generated by ICU bedside monitors. The novelty in our approach lies in the extraction of 216 relevant features to capture the characteristics of all alarms, from both arterial blood pressure (ABP) and electrocardiogram (ECG) signals. Our multivariate approach mitigates the imprecision caused by existing heartbeat/peak detection algorithms. Unlike existing methods on ICU false alarm detection, our approach does not require separate techniques for different types of alarms. The experimental results show that our approach can achieve high accuracy on false alarm detection, and can be generalized for different types of alarms.

I. INTRODUCTION

Automated electronic monitoring systems in intensive care units (ICU) routinely collect vast amounts of real-time patient vital signs data, including blood pressure and electrocardiogram (ECG), via bedside monitors. If abnormalities in the vital signs are detected (e.g. if they fall outside the range of some pre-set thresholds), the monitoring system triggers the alarm and notifies the nurses or physicians in charge. However, the ICU monitoring systems often have a high level of sensitivity in detecting patients’ abnormal status. While it is important to not miss any important alarms, the sensitivity leads to a high false alarm rate, resulting in a common phenomenon known as alarm fatigue. According to Lawless [1], ICU false alarm (FA) rates are as high as 86%. In fact, only 2% to 9% of alarms have been found to be important for patient management [2]. The overwhelming frequency of alarms, particularly the false alarms, have negative impact on patients and hospital staff [3]. More specifically, alarm fatigue for medical staff may lead to longer response time or missing of important alarms. In addition, the work by Gabor et al. [4] shows that noises from ICUs, e.g. noises made by the machines, could lead to slower recovery for the patients. Therefore, the reduction of false alarms in the ICU is an important problem that has gained much popularity in recent decades.

A variety of techniques have been developed for the problem of false alarm suppression over the past few years. Sayadi et al. [5] present a nonlinear joint dynamical model based on Extend Kalman Filter (EKF) to enhance the performance life-threatening alarms detection. However, the model works only when all channels of signals are available, and the cost of computation is high. Salas-Boni et al. [6] introduce a wavelet transform based method to detect false ventricular tachycardia alarms but they only consider one type of life-threatening alarms. Behar et al. [7] introduce a method based on signal quality generated from ECG segments for false alarm detection. Aboukhalil et. al. [8] introduce an algorithm to reduce the false critical arrhythmia alarms using morphological and timing information derived from the arterial blood pressure (ABP) signal. The proposed algorithms are data-specific, in the sense that each type of alarm is handled differently, based on the definition of the alarm. Baumgartner et al. [9] use a data mining process to detect false alarms. However, the authors only consider instances that contain the second Einthoven ECG derivation and the ABP signal. The majority of alarm records in the well-known MIMIC II (Multiparameter Intelligent Monitoring in Intensive Care) database [10] does not have these two signals simultaneously.

In this work we address the problem of false alarm suppression on five types of life threatening ICU alarms, in order to alleviate the alarm fatigue problem. Existing work on ICU alarm reduction either are data-specific i.e., they require learning of specific parameters or features based on alarm definitions, or focus only on small subset of data that meet certain requirements. Our technique learns from the same set of features regardless of alarm definition, and consists of the following three stages:

1) Feature extraction: The MIMIC II database [10] contains data for five types of life threatening alarms: (i) Asystole, (ii) Extreme Bradycardia (EB), (iii) Extreme tachycardia (ET), (iv) Ventricular tachycardia (VT) and (v) Ventricular fibrillation (VT/VF). Each record in the database is annotated with the alarm type and the time stamp of the alarm, and each record contains the Arterial Blood Pressure (ABP) and Electrocardiogram (ECG) signals. Using the ABP and ECG signals, our goal is to classify an alarm into True (alarm) or False (alarm). Due to the high level of noise and the occasionally unstable voltage of bedside monitors, we find that existing anomaly detection techniques that focus on discovering patterns directly from the raw time series data typically yield poor results.

Therefore, instead of using the original ABP/ECG signals directly, we introduce a feature extraction step to transform the raw time series data into a new features space. The features extracted from the original ABP and ECG signals can potentially be used to distinguish false alarms from true alarms. The new features are generated by extracting relevant statistical information from the original signals.
2) Feature selection: For all the alarm types, our algorithm extracts the same set of features. However, for different types of alarms, the relevant or important features may vary. For example, the Asystole alarms are triggered by a default asystolic pause of 4 seconds. In other words, features related to the largest heartbeat interval in the ABP and ECG signals are more important than other features. However, for Extreme tachycardia, a condition where the Heart Rate exceeds 140 beats per minute (BPM), the information about the largest interval is not as informative as the features related to heart rates (or small intervals which correspond to fast heart rates). In addition, correlation may exist among features, which may impact the accuracy of false alarm detection. Therefore, we incorporate feature selection algorithms in our technique, which selects the key features for different types of alarms. Differing from existing techniques [8], our unified framework does not require separate learning techniques for different types of alarms.

3) Classification: In this stage, our algorithm trains a classifier using historical alarm data. The false alarm suppression problem is formulated as a binary classification problem that outputs one of two class labels: True or False alarm.

In summary, we make the following contributions in this work: (i) Our method detects false alarms without any prior knowledge of the alarms. (ii) By transforming raw ABP/ECG signals into other feature space, we achieve better accuracy on false alarm detection compared to existing work.

The remainder of the paper is organized as follows: we discuss our methodology in Section II. Section III shows experimental validation of our technique on the MIMIC II database [10]. We conclude in Section IV and offer suggestions for future work.

II. METHODS

Our approach consists of three components: feature extraction, feature selection, and classification. Prior to feature extraction, we pre-process the data by removing signals that do not contain sufficient information about patient condition. For each alarm, we extract the appropriate segment in the signal surrounding the alarm. For feature extraction, we transform the ABP and ECG signals into the appropriate feature space that removes noise and helps detect false alarms. After extracting the features from the signals, we obtain a large set of features; however, these features have different significance for different alarm types. In this phase the algorithm automatically identifies the relevant feature set for each alarm type via feature selection. Once the relevant features are selected, we train a binary classifier that outputs one of the two class labels (True or False alarm). The alarms that are classified as False alarms in the test set should be suppressed.

A. Data Source and Preprocessing

The dataset we use in this work is from the MIMIC II (Multiparameter Intelligent Monitoring in Intensive Care) Waveform Database from Physionet [11]. As in prior studies, a subset of records from the MIMIC II database were selected with two criteria [8]: (i) a critical ECG arrhythmia alarm was issued at some time during the ICU stay, and (ii) one channel of ECG and an ABP waveform were present at the time of the arrhythmia alarm.

There are five types of life-threatening ECG arrhythmia alarms present in the MIMIC II database. According to ANSI/AAMI-EC13 Cardiac Monitor Standards [12], they are defined as: (i) Asystole: asystolic pause of 4 seconds, as shown in Figure 1; (ii) Extreme Bradyarrhythmia (EB): heart rate less than 40 beats per minute (BPM), as shown in Figure 2; (iii) Extreme tachycardia (ET): heart rate greater than 140 BPM, as shown in Figure 1; (iv) Ventricular tachycardia (VT): five continuous ventricular heart beats, as shown in Figure 1; and (v) Ventricular fibrillation (VT/VF): a fibrillatory waveform lasting for more than 4 second, as shown in Figure 1.

There are five types of life-threatening ECG arrhythmia alarms present in the MIMIC II database. According to ANSI/AAMI-EC13 Cardiac Monitor Standards [12], they are defined as: (i) Asystole: asystolic pause of 4 seconds, as shown in Figure 2; (ii) Extreme Bradyarrhythmia (EB): heart rate less than 40 beats per minute (BPM), as shown in Figure 1; (iii) Extreme tachycardia (ET): heart rate greater than 140 BPM, as shown in Figure 1; (iv) Ventricular tachycardia (VT): five continuous ventricular heart beats, as shown in Figure 1; and (v) Ventricular fibrillation (VT/VF): a fibrillatory waveform lasting for more than 4 second, as shown in Figure 1.

According to the AAMI standards [12], the asystole and rate-limit arrhythmia alarms must be triggered within 10 seconds of the onset of the event. In this work, for each alarm we take a 17-seconds (13 seconds prior to the alarm onset and 4 seconds after the alarm) corresponding ABP and ECG waveform segments as the input data.

We define a record or an instance as a pair of corresponding 17-second ABP signal and 17-second ECG signal of an annotated alarm. Figure 2 is an example of an alarm instance (a record). The top figure is a 17-second ABP signal and the bottom figure is a 17-second ECG signal; the vertical red dashed lines denote where the annotated alarm is.

In the pre-processing step, we remove signals of poor quality, since those signals lack information on patient condition. Figure 3 shows such an example of an ABP signal, whose...
low values may be caused by loss of connection. Based on the work in [13], if the range of an ABP signal is less than 20, it is of poor quality and is discarded in this study.

Fig. 3. Example of a poor quality ABP signal. The signal is nearly flat.

B. Feature Extraction

Since all of the alarms are related to some abnormality in the heart rate, we extract relevant features from the raw ECG and ABP signals as described below. We categorize the features into three groups: whole-segment-based features (for both ECG and ABP, independently), interval-based features (for both ECG and ABP, independently), and features that combine both ECG and ABP signals.

1) Whole segment based features: Whole-segment-based features are statistical properties computed from the whole segment of the ABP or ECG signal using the peak detection of the ABP or ECG signal.

- **Mean Value**: The average value of the segment of the ABP or ECG signal.
- **Maximum Value**: The maximum value of the segment of the ABP or ECG signal.
- **Minimum Value**: The minimum value of the segment of the ABP or ECG signal.
- **Range value within an interval**: The difference between the maximum value and the minimum value within a detected interval.
- **Mean Interval between Peak**: The average length of intervals between peaks. Each of these statistical properties is computed from the whole 17-second segment. We define the features as follows:

\[
\text{Mean Value} = \frac{1}{|p|} \sum_{i=2}^{m} p_i - p_{i-1} \quad (1)
\]

where each \( p_i \) denotes the location of the peak in a heartbeat, and \( m \) is the number of peaks.

In summary, for each alarm instance, we extract a total of 2 * 7 = 14 whole-segment-based features (7 features for ABP and 7 features for ECG).

2) Interval based features: The interval-based features are statistical measures computed from each heartbeat in an instance. The individual heartbeats are detected by the algorithm described in [14]. Suppose the heartbeat/peak detection algorithm for signal \( S \) outputs a series \{\( p_i, \ i = 1 \ldots m \)\} where \( m \) is the number of detected heartbeats in a record, and \( p_i \) is the location of the \( i^{th} \) peak. We can define intervals as the segments \{\( N_j, j = 1 \ldots (m-1) \)\}, where \( N_j \) is the section between two consecutive peaks \( p_j \) and \( p_{j+1}, j = 1 \ldots (m-1) \).

For the problem of false alarm detection, we are only interested in intervals that are potentially indicative of abnormalities, e.g. the largest or the smallest interval from each instance. Taking into consideration the potential imperfection of the peak detection algorithm due to noise, we take the top-\( K \) (\( K \in 1 \ldots 5 \)) largest and smallest intervals, respectively. For each of the top-\( K \) largest and smallest intervals, we extract the following interval-based features:

- **Minimum value within an interval**: The minimum value within a detected interval \( N_j \).
- **Maximum value within a interval**: The maximum value within a detected interval \( N_j \).
- **Mean value**: Average value within a detected interval.
- **Standard deviation**: The standard deviation within a detected interval.
- **Range value within an interval**: The difference between the maximum value and the minimum value within a detected interval.
- **Position difference of maximum and minimum**: The position difference between the maximum point and the minimum point within a detected interval.

\[
pd_{Itv} = (|\text{argmax}_{x \in N_j} x - \text{argmin}_{x \in N_j} x|) \quad (2)
\]

- **Interval Length**: Length of a detected interval.
- **Area**: Area under the signal of a detected interval.
- **Maximum change**: Maximum change within a detected interval.

\[
mc_{Itv} = \max_{x_i \in N_j} (|x_{i+1} - x_i|) \quad (3)
\]

For each interval \( N_j \) in a signal \( S \) (ECG or ABP), there are a set of nine statistical measures as described above. Since the features are extracted from both the ABP and the ECG signals, we will have a total of 2 * 9 = 18 features per record (i.e. per alarm instance). Also we take the top-\( K \) (\( K \in 1 \ldots 5 \)) largest and smallest intervals, respectively, and average the values for each feature from the \( K \) intervals. Instead of setting a fixed value for \( K \), we consider all \( K \) from 1 to 5, resulting in 18 * 2 * 5 = 180 features for each record (here we multiply by 2 because we consider both largest and smallest intervals).

The reason we introduce the features above instead of merely extracting the heart rate (HR) (as [8] did for alarms Asystole and Extreme bradycardia) is that it cannot be generalized to other types of alarms. For more complicated alarm conditions, possibly those not defined in this database, one would need to extract other alarm-specific features. While the
features we extract are largely motivated by the current alarm definitions, by specifying a wide range of features, we can learn the best features automatically based on the training data without knowledge of specific alarms.

**Consecutive intervals:** In addition to the features mentioned above for single interval, information from consecutive heartbeats can provide important insight as well. Therefore, we also consider the information related to $c = 2 \ldots 4$ consecutive intervals. Again, we consider the largest and the smallest $c$ consecutive interval for ABP and ECG. In this step, we generate $3 \times 2 \times 2 = 12$ features (i.e. 3 choices for $c$, 2 choices for signal type ABP/ECG, and 2 choices for smallest/largest).

The largest $c$ consecutive intervals for ABP or ECG is defined as the segment between $p_i$ and $p_i-c$, where $(p_i - p_i-c) > (p_j - p_j-c), \forall j \neq i, j = (c+1) \ldots m$. Similarly, the smallest $c$ consecutive intervals for ABP or ECG is defined as the segment between $p_i$ and $p_i-c$, where $(p_i - p_i-c) < (p_j - p_j-c), \forall j \neq i, j = (c+1) \ldots m$.

3) **Features from combination of ABP and ECG:** So far, all the features described are extracted from ABP and ECG independently. However, the peak detection algorithm may not detect all peaks/heartbeats correctly due to the noise in the signals. For example, there may be missing peaks which may result in erroneously long intervals. In order to overcome this problem, we introduce the combined features which use the peaks detected in one signal to improve the accuracy of peaks detected from the other signal. Considering the chance of missing peaks in both the ABP and ECG signals simultaneously is low, the overall accuracy can be improved.

Suppose $p_{a,j}$ and $p_{c,j}$ are the $j^{th}$ peak found by the peak detection algorithm in the ABP and ECG signals, respectively. Then we can define the $j^{th}$ interval in ECG (with the help of ABP signal) as:

$$i_{v_{e,j}} = \max_{p_{a,k} < p_{a,k+1} < p_{a,k+1} < p_{e,j+1}} (p_{a,k+1} - p_{a,k}) \quad (4)$$

where $k$ could be any value in $\{1 \ldots m - 1\}$. Eq. (4) states that if at least one complete ABP interval exists between an ECG interval $[p_{e,j}, p_{e,j+1}]$, then the largest ABP interval between $p_{c,j}$ and $p_{e,j+1}$ is used as the $j^{th}$ ECG interval length. If there is no complete ABP interval between $p_{c,j}$ and $p_{e,j+1}$, then $i_{v_{e,j}}$ is just $p_{e,j+1} - p_{e,j}$. Similarly, the $i_{v_{a,j}}$ in ABP (with the help of ECG signal) could be defined the same way. Similar to the features generated from single signal, we compute the average of the top $K$ largest intervals. We set $K \in 1 \ldots 5$, so the number of features (i.e. average interval lengths) generated here is $5 \times 2 = 10$ (5 from ABP, 5 from ECG).

In summary, the total number of features (whole-segment-based, interval-based, and combination) we have is $14 + 180 = 12 + 10 = 216$.

**C. Feature Selection Algorithms**

The correlations among the features may impact the performance of the classifier and result in biased outcome or overfitting. In addition, as the number of features increases, the data may become under-sampled — a phenomenon known as the curse of dimensionality. In this phase, we perform feature selection to automatically learn the best features suitable for each alarm type. We consider several different feature selection approaches: principal component analysis (PCA), decision tree, bagged decision tree, and rank features by class separability.

PCA finds a subset of linear combinations of the original features called the principal components [15]. Without much information loss, the new features capture the covariance structure of the original features in reduced space.

Decision tree is a classic machine learning tool for classification and regression. By placing the feature that best distinguishes the data (i.e. the one resulting in the highest information gain) at the root, a decision tree induction process recursively partitions the data samples into subsets of samples based on the feature splitting criterion. Since a decision tree typically contains only a subset of (most informative) features, we can also use it for feature filtering or feature selection — the features contained in the tree will be selected as input for a classifier such as SVM.

Bagged decision tree is similar to decision tree. The difference is that the bagged decision tree does not take the whole training data to build a predictive model. It trains a set of decision trees by generating in-bag samples by oversampling classes with large misclassification cost and undersampling classes with small misclassification cost. Every feature gets a score based on its performance in decision tree. Features with high score will be selected.

The ranking method ranks features by their class separability, using their scores measured by T-Test[16] and F-score[17] criteria. Top ranked features will be selected as input for the classification phase.

**D. Classification**

In this section, we use the features selected from Section II-C for classification. We classify a given alarm into True or False alarm. We use Support Vector Machine (SVM) as our main choice of classifier since it is one of the most widely used classifiers. The SVM implementation we use is LibSVM [18]. Other classification methods that we consider in our experiments include decision tree (DT), bagged decision (BDT), and k nearest neighbor (KNN).

**E. Evaluation Metrics**

We evaluate the classifier performance based on three metrics: classification accuracy; false alarm (FA) suppression rate; and true alarm (TA) suppression rate. Classification accuracy measures the percentage of alarms that are correctly classified. False alarm suppression rate measures the percentage of false alarms correctly identified. True alarm suppression rate measures the percentage of true alarms incorrectly classified as false alarms. Obviously, we want to maximize the false alarm suppression rate and minimize the true alarm suppression rate (down to zero, ideally).
III. RESULTS AND DISCUSSION

A. Experimental Design

We implement all of our algorithms in Matlab. The dataset we use are from PhysioNet’s MIMIC II Waveform Database [10]. There are a total of 5,569 alarms in the dataset after removing records with insufficient or erroneous signals. Out of the 5,569 alarms, 3,133 of them are true alarms and 2,256 are false alarms. Recall there are five types of life-threatening arrhythmia alarms in the database. The number of instances for each type of alarm is shown in Table I.

<table>
<thead>
<tr>
<th>Type</th>
<th>Num of True Alarm</th>
<th>Num of False Alarm</th>
<th>Total Alarm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asystole</td>
<td>34</td>
<td>515</td>
<td>549</td>
</tr>
<tr>
<td>EB</td>
<td>498</td>
<td>208</td>
<td>706</td>
</tr>
<tr>
<td>ET</td>
<td>1560</td>
<td>416</td>
<td>1976</td>
</tr>
<tr>
<td>VT</td>
<td>1132</td>
<td>893</td>
<td>2025</td>
</tr>
<tr>
<td>VT/VF</td>
<td>49</td>
<td>224</td>
<td>313</td>
</tr>
</tbody>
</table>

B. Results

The reason for using extracted features instead of using ABP/ECG signals directly is two folds: (i) We obtain important information related to the characteristics of alarms. For example, as shown in Figure 4(a), the signal that contains the true alarm is very similar to the one that contains false alarm. Thus, using the raw time series directly will lead to poor performance. Using the extracted features, in particular the largest/smallest $K$ intervals, we can separate the two classes and detect this alarm easily. (ii) The signal-to-noise ratio in ABP/ECG signals leads to erroneous distance computations between time series. Converting the noisy signal into the proposed feature space mitigates this problem. As shown in Figure 4(b), both ECG signals contain a false alarm, but they look dissimilar to each other. Transforming them to the proposed feature space shows that both have similar intervals.

The performance of our algorithms is evaluated using the criteria described in Section II-E. Due to the limited data size (in particular, the Asystole alarm since it is highly unbalanced), we use 5-fold cross validation to evaluate our methods. We first compare our method with other methods in [8], [9], and then compare results from different combinations of feature selection methods (no selection, decision tree/bagged decision tree, and ranking features) and classification methods (SVM, decision tree, bagged decision tree and KNN).

1) Comparison with other methods: We compare our approach, Feature-based False Alarm Detection (FFAD), with prior work for false alarm detection [8], [9]. Since we do not have the same datasets used in [8] or in [9], we try our best effort to implement and run their methods on our dataset for comparison. The results are summarized in Table II. The bold items in the table indicate the best value for each method.

Comparing to the best previous work, in general, our method has higher classification accuracy by 7.16%, higher false alarm suppression rate by 13.96%, and lower true alarm suppression rate by 0.99%. For all types of alarms, our method wins on false alarm detection rate and classification accuracy. For true alarm detection, our method does not always win, but the loss is small: comparing to the best previous work, we only have 1 (Asys.), 8 (EB), and 3 (VT/VF) more instances that are misclassified.

2) Comparison of different feature selection and classification algorithms: In this section we compare different combinations of feature selection and classification algorithms. The results are summarized in Table III. The bold items in the table indicate the best value for each type of alarm. According to the results, complex classifiers such as SVM and BDT have better performance on the proposed features. However, the proposed features could be used in conjunction with any feature selection and classification methods.

IV. CONCLUSION AND FUTURE WORK

In this work, we introduce a novel technique to detect false critical ECG arrhythmia alarms. We extract three different types of features from the original ECG/ABP signals to capture the characteristics of alarms. Our algorithm automatically selects relevant (multivariate) features for different types of
Table III. Classification results (5-fold cross validation) from different combinations of classification (CLS.) algorithms and feature selection (FS) algorithms.

<table>
<thead>
<tr>
<th>C1s. Alg</th>
<th>FS Alg</th>
<th>TP</th>
<th>FN</th>
<th>FP</th>
<th>TN</th>
<th>Acc. (%)</th>
<th>Suppression Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>None</td>
<td>3109 204 390 1866 89.33</td>
<td>6.16</td>
<td>82.71</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VT</td>
<td>None</td>
<td>2923 390 395 1861 85.90</td>
<td>11.77</td>
<td>82.49</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VT/VVF</td>
<td>None</td>
<td>2975 338 345 1911 87.74</td>
<td>10.20</td>
<td>84.71</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EB</td>
<td>None</td>
<td>3010 303 262 1994 88.95</td>
<td>9.15</td>
<td>88.39</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ET</td>
<td>None</td>
<td>2975 338 345 1911 87.74</td>
<td>10.20</td>
<td>84.71</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BDT None</td>
<td>28     6</td>
<td>3</td>
<td>512</td>
<td>98.36</td>
<td>17.65</td>
<td>99.42</td>
<td></td>
</tr>
<tr>
<td>DT None</td>
<td>27     7</td>
<td>6</td>
<td>509</td>
<td>97.63</td>
<td>20.59</td>
<td>98.83</td>
<td></td>
</tr>
<tr>
<td>KNN None</td>
<td>30     4</td>
<td>8</td>
<td>507</td>
<td>97.81</td>
<td>11.76</td>
<td>98.45</td>
<td></td>
</tr>
<tr>
<td>SVM None</td>
<td>3      1</td>
<td>9</td>
<td>506</td>
<td>98.18</td>
<td>2.94</td>
<td>98.25</td>
<td></td>
</tr>
<tr>
<td>SVM DT</td>
<td>29     5</td>
<td>4</td>
<td>511</td>
<td>98.36</td>
<td>14.71</td>
<td>99.22</td>
<td></td>
</tr>
<tr>
<td>SVM BDT</td>
<td>31     3</td>
<td>3</td>
<td>512</td>
<td>98.91</td>
<td>8.82</td>
<td>99.42</td>
<td></td>
</tr>
<tr>
<td>SVM TTest</td>
<td>32     2</td>
<td>2</td>
<td>513</td>
<td>99.27</td>
<td>5.88</td>
<td>99.61</td>
<td></td>
</tr>
<tr>
<td>SVM Fscore</td>
<td>27     7</td>
<td>4</td>
<td>511</td>
<td>98.00</td>
<td>20.59</td>
<td>99.22</td>
<td></td>
</tr>
<tr>
<td>SVM PCA</td>
<td>16     18</td>
<td>2</td>
<td>513</td>
<td>96.36</td>
<td>52.94</td>
<td>99.61</td>
<td></td>
</tr>
</tbody>
</table>

Acknowledgment

This research is partially supported by the National Science Foundation under Grant No. 1218325 and Grant No. 1252318, and the Naval Research Lab. We would like to thank Daniel McFarlane and Larry Nielsen for their helpful advice.

References


Alarms by feature selection. The experimental results show that our approach outperforms other methods in most cases. For future work, we will focus on further reducing the true alarm suppression rate, e.g. via cost-sensitive learning.