Feature Selection Technique Using Ant Colony Optimization on Keystroke Dynamics

Shivam Sharma

Abstract: The work is concerned with the use of Ant Colony Optimization algorithm for feature selection of Keystrokes Dynamics and comparison of classification accuracy of Multi-SVM and KNN classifiers. There are various approaches used for feature subset selection but, ACO algorithm gives good performance than other feature selection algorithm like Genetic Based algorithm and Particle Swarm Optimization. In this, first all features are extracted from benchmark dataset, then Multi-SVM and KNN classifiers are trained using all features and their classification accuracy is compared with same training set and test set .Then, features are reduced by Ant Colony Optimization algorithm and then Multi-SVM classifier is trained using reduced features and performance is compared before and after feature selection. The study deals with the use of this technology in ID Password authentication in computer systems, Mail Service Provider and where ID Password is used.

Keywords: Ant Colony Optimization, FAR, Feature Subset Selection, FRR, Keystroke Dynamics, K-Nearest Neighbour, Pheromone, Support Vector Machine.

1. INTRODUCTION:

Securing the sensitive data and computer systems by allowing ease access to authenticated users and withstanding the attacks of imposters is one of the major challenges in the field of computer security. ID and password are the most widely used method for authenticating the computer systems. But, this method has many loop holes such as password sharing, shoulder surfing, brute force attack, dictionary attack, guessing, phishing and many more. Keystroke Dynamics is one of the famous and inexpensive behavioural biometric technologies, which identifies the authenticity of a user when the user is working via a keyboard. Keystroke dynamics is the process of analysing the way a user types at a terminal by monitoring the keyboard inputs thousands of times per second in an attempt to identify users based on habitual typing rhythm patterns. Keystroke Dynamics is a relatively new method of biometric identification. It is assumed as a robust behavioural biometric. The functionality of this biometric is to measure the dwell time and flight time for changing keyboard actions. In typing a phrase or a string of characters, the typing dynamics or timing pattern can be measured and used for identity verification. More specifically, a timing vector consists of the keystroke duration times interleaved with the keystroke interval times at the accuracy of milliseconds (ms). During the feature extraction phase, user keystroke features from one's name or password are captured, processed and stored in a reference file as prototypes for future use by system in subsequent authentication operations. During the verification phase user keystroke features are captured, processed in order to render an authentication decision based on the outcome of a classification process of the newly presented feature to the pre stored. It would be necessary for the user to type his/her name or password a number of times in order for the system to be able to extract the relevant features that uniquely represent the user. However, the task of typing one's name or password over and over is both tiring and tedious in the feature extraction phase, which could lead users to alter their normal typing pattern. Thus, most systems based on biometrics are required to work with a summarized set of information from which to extract knowledge. In order to reduce this problem, we could eliminate some features of the original dataset, selecting only the best ones. This is called Feature Selection. Feature subset selection is applied to high dimensional data prior to classification. Feature subset selection is essentially an optimization problem, which involves searching the space of possible features to identify one that is optimum or near-optimal with respect to certain performance measures, since the aim is to obtain any subset that minimizes a particular measure. It is the technique of selecting a subset of relevant features for building robust learning models. It provides better understanding of the data by selecting important features within the data. The central assumption when using a feature selection technique is that the data contains many redundant or irrelevant features. Redundant features are those which provide no more information than the currently selected features, and irrelevant features provide no useful information in any context. In this, Ant Colony Optimization algorithm is proposed for feature selection of keystrokes. In order to train the system, we need a classifier

ISSN 2229-5518

that can efficiently identify which class a particular user belongs to. And for that, we need classifier. Since there are multiple classifiers, how do we decide what classifier should be used. Therefore we need to compare the efficiency of the classifiers in order to decide which one is better. Apart from that, since we have many features from each user's entry, it makes the classification process so moderate that it takes hours to decide which class a user belongs to. So we need to decide some important features that are well enough to decide the class of the user without taking much time and maintain the accuracy on the other hand as well. And therefore, here we need to introduce a feature selection technique that will efficiently select some main features that will help to decide the class of a user which he belongs to. The fact that computers regularly store private, sensitive and classified information makes it very important that we can confidently identifies its users. Traditionally this has been achieved through password authentication systems. However, these systems are far from perfect. For instance, if a password becomes compromised it is no longer adequate for authenticating its rightful owner. So far in this field the most promising techniques focus on patterns in the timing of a user's typing. This is called as biometric keystroke authentication'. Compared to more conventional biometrics such as fingerprint or iris authentication, it offers the following advantages:

- It does not require special tools or hardware, only a conventional keyboard.
- It is non-invasive for the user.
- It can easily be deployed in conjunction with existing authentication systems.
- It can be collected without the user's knowledge.
- It has reduced computation time
- It has improved accuracy
- It has better false acceptance rate

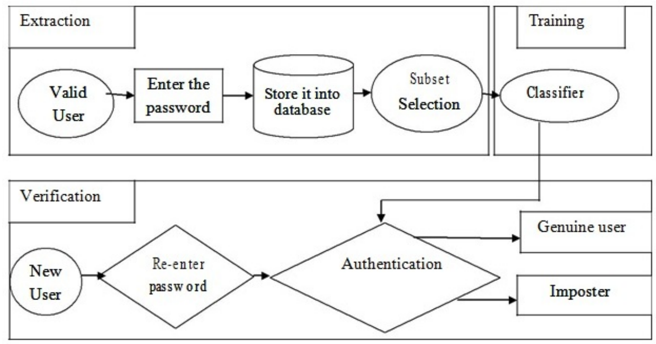
2. OBJECTIVE

- To compare the Multi-SVM and KNN classifiers.
- To test the accuracy of feature selection technique using Ant Colony Optimization on Keystroke Dynamics - Benchmark dataset.
- To reduce the curse of dimensionality.
- To prevent from over fitting.

3. SCOPE

- The dataset is divided into training and testing part.
- Performance is measured using SVM classifier.
- ACO feature selection technique is applied on the given dataset to reduce the features involved.
- Performance is measured using SVM classifier of reduced feature subset.
- Compare the performance before and after applying feature selection technique.
- Build GUI to implement above things.

4. KEYSTROKES DYNAMICS ANALYSIS FRAMEWORK



During the feature extraction phase user keystroke features from one's name or password are captured, processed and stored in a reference file as prototypes for future use by system in subsequent authentication operations. During the verification phase user keystroke features are captured, processed in order to render an authentication decision based on the outcome of a classification process of the newly presented feature to the pre-stored prototypes .It would be necessary for the user to type his/her name or password a number of times in order for the system to be able to extract the relevant features that uniquely represent the user. However, the task of typing one's name or password over and over is both tiring and tedious in the feature extraction phase, which could lead users to alter their normal typing pattern. Thus, most systems based on biometrics are required to work with a summarized set of information from which to extract knowledge. In order to reduce this problem, we could eliminate some features of the original dataset, selecting only the best ones in terms of class cohesion.

5. FEATURES IN KEYSTROKES DYNAMICS

5.1. DWELL TIME OR DURATION (DT) - Dwell time refers to how long a key is pressed until it is released. It is the difference between time of pressing a key and releasing the same key and can be calculated as DT = R1 – P1 where R1 indicates a key release and P1 indicates a key press.

5.2. FLIGHT TIME (FT) - Latencies or Flight time is the interval time between key press and release of different keys.

5.3. DIGRAPH (DI) - Digraph is the Elapsed time between the first key press and the second key press. It is given as follows: DI = P2 − P1 where P2 is the pressing of second key and P1 is the pressing of first key.

6. FEATURE SUBSET SELECTION

Feature selection is the process of selecting a subset of the terms occurring in the training set and using only this subset as features for further processing. It is the process of selecting a subset of 6 relevant features for use in model construction. The central assumption when using a feature selection technique is that the data contains many redundant or irrelevant features. Redundant features are those which provide no more information than the currently selected features, and irrelevant features provide no useful information in any context. Feature selection techniques are a subset of the more general field of feature extraction. Feature selection techniques provide three main benefits when constructing predictive models:

1. Improved model interpretability.
2. Shorter training times.
3. Enhanced generalization by reducing over fitting.

6.1 THREE TYPES OF FEATURE SELECTION TECHNIQUES

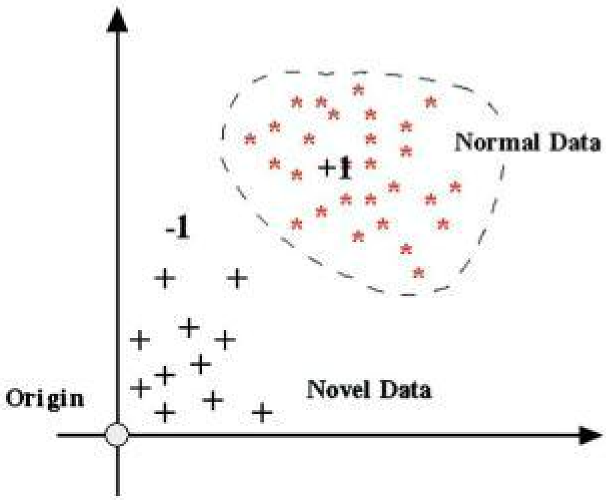
6.1.1. Wrapper - Wrapper methods use a predictive model to score feature subsets. Each new subset is used to train a model, which is tested on a hold-out set. Counting the number of mistakes made on that hold-out set (the error rate of the model) gives the score for that subset. As wrapper methods train a new model for each subset, they are very computationally intensive, but usually provide the best performing feature set for that particular type of model.

6.1.2. Filter- Filter methods use a proxy measure instead of the error rate to score a feature subset. This measure is chosen to be fast to compute, whilst still capturing the usefulness of the feature set. Common measures include the mutual information, the point wise mutual information, Pearson product-moment correlation coefficient, inter/intra class distance or the scores of significance tests for each class/feature combinations. Filters are usually less computationally intensive than wrappers, but they produce a feature set which is not tuned to a specific type of predictive model.

6.1.3. Embedded- Embedded methods are a catch-all group of techniques which perform feature selection as part of the model construction process. The exemplar of this approach is the LASSO method for constructing a linear model, which penalises the regression coefficients, shrinking many of them to zero. Any features which have non-zero regression coefficients are 'selected' by the LASSO algorithm.

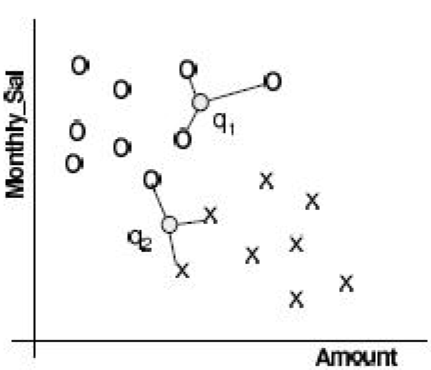
7. SUPPORT VECTOR MACHINE CLASSIFIER

Support Vector Machine (SVM) is a supervised learning model with associated learning algorithms that analyse data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, a SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. Yu and Cho [6] proposed an algorithm where support vector machine is used as classifier. It gained great interests in pattern recognition. One reason is its theoretical foundation of structural risk minimization principle minimizing time consuming trial- and-error search of hyper parameters, which is ubiquitous in the neural network model. Another reason for its success is its excellent classification performance in numerous application areas. The idea is to map the data into the feature space corresponding to the kernel, and then to separate them from the origin with a maximum margin. The algorithm returns a decision function f that takes the value C1 in a small region capturing most of the normal data, and -1 elsewhere. For a new point x, the value f(x) is determined by evaluating which side of the hyper plane it falls on, in feature space. Liu et al. [10] presented approach of One-Against-All Multi-Class SVM Classification Using Reliability Measures. Support Vector Machines (SVM) is originally designed for binary classification. The conventional way to extend it to multi-class scenario is to decompose an M- 9 class problem into a series of two-class problems, for which one-against-all is the earliest and one of the most widely used implementations. One drawback of this method, however, is that when the results from the multiple classifiers are combined for the final decision, the outputs of the decision functions are directly compared without considering the competence of the classifiers.



8. K-NEAREST NEIGHBOUR CLASSIFIER

K-Nearest Neighbours algorithm is a non-parametric method used for classification. Input consists of the k closest training examples in the feature space. In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbour. K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms. Cunningham et al. [11] explained KNN in their paper. K-NN is very simple to understand and easy to implement. This paper presented an overview of techniques for Nearest Neighbours 10 classification focusing on; mechanisms for assessing similarity (distance), computational issues in identifying nearest neighbours and mechanisms for reducing the dimension of the data.



9. ANT COLONY OPTIMIZATION (ACO)

Ant algorithm was first proposed as a multi-agent approach to difficult combinatorial optimization problems such as the Traveling Salesman Problem (TSP) and the Quadratic Assignment Problem (QAP). There are currently various activities in the scientific community to extend and apply ant-based algorithms to many different discrete optimization problems. The ACO heuristic has been inspired by the observation on real ant colony's foraging behaviour, and that ants can often find the shortest path between food source and their nest. Ant individuals transmit information through the volatile chemical substances which ants leave in its passing path known as the —pheromone‖ and then reach the purpose of finding the best way to search food sources. An ant encountering a previously laid trail can detect the dense of pheromone trail. It decides with high probability to follow a shortest path, and reinforce that trail with its own pheromone. The large amount of pheromone is on the particular path, the large probability is that an ant selects that path and the paths pheromone trail will become denser. At last, the ant colony collectively marks the shortest path, which has the largest pheromone amount. Such simple indirect communication way among ants embodies actually a kind of collective learning mechanism.

10. SYSTEM DESIGN

10.1 SYSTEM ARCHITECTURE

The given flow chart describes the overall architecture of a system. Firstly features are extracted from the Keystroke Benchmark dataset. Then these features are analysed by the SVM classifier. All features are evaluated by Multi-SVM classifier and KNN classifier. Performance is compared of both classifiers. After that, numbers of feature are reduced by the ACO algorithm. Then the reduced feature subsets are evaluated by the SVM classifier. Performance is compared of dataset before applying feature selection algorithm and after applying feature selection algorithm using Multi-SVM classifier. And based on above result accuracy is determined.

ISSN 2229-5518

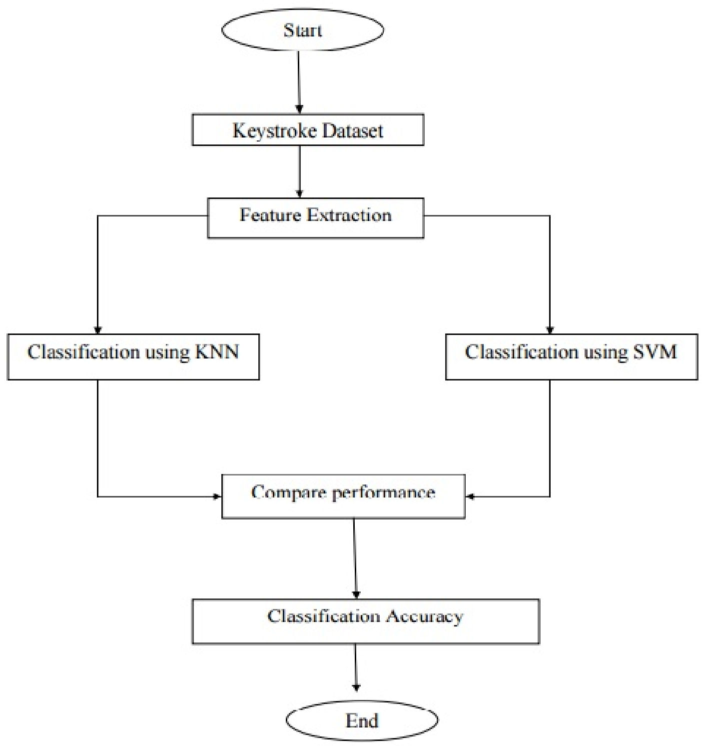


Figure: Comparison of SVM and KNN classifiers,

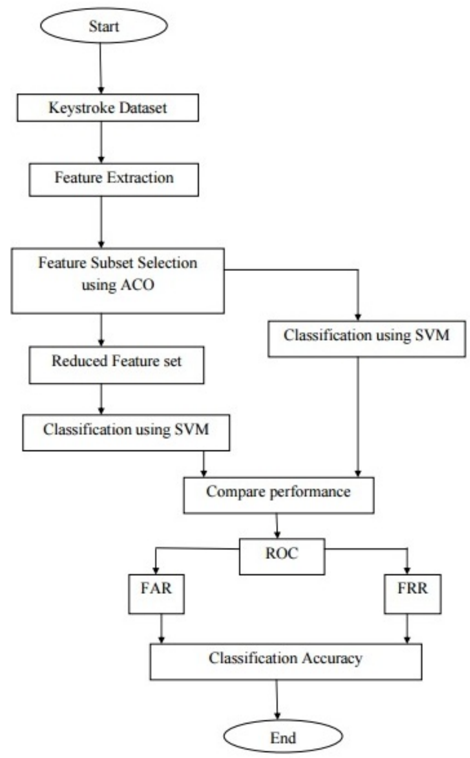


Figure: System Architecture

ISSN 2229-5518

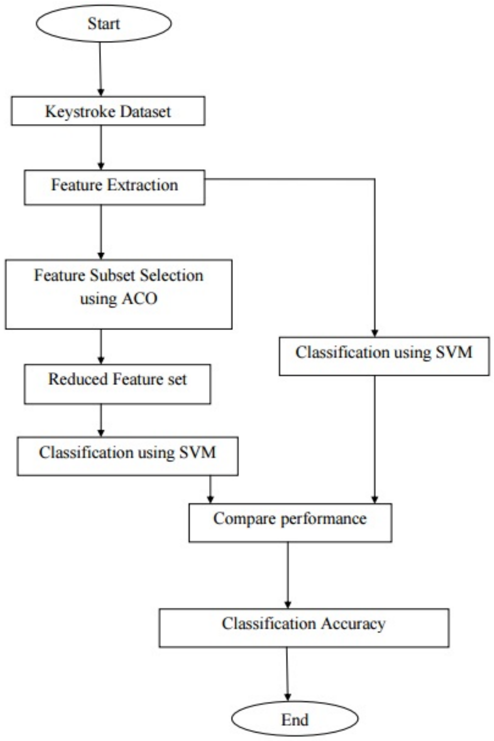


Figure: Comparison of SVM before and after feature selection

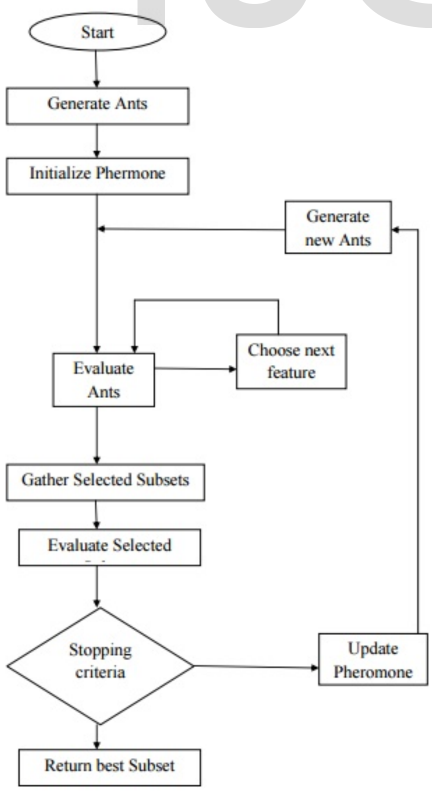


Figure: Ant Colony Optimization feature selection

10.2 ACO AS FEATURE SELECTION ALGORITHM

The main idea of the proposed approach is to provide a fully connected N*N graph (where N is the total number of the attributes or features present in the dataset). The graph behaves like a search space for the ants to move, where links represent the connection between features of a particular dataset and nodes are the features. Each ant constructs a candidate solution in this search space by traversing a path of nodes and links. This path is actually the subset of the features. After an ant has completed its tour, the fitness of the traversed path (selected features) is calculated for the selected features and then checking the accuracy of the learned model. The average accuracy is the fitness of that particular feature subset and is used to update the pheromone values. This process continues until a stopping criterion is met. After termination of the algorithm the features set that has the best accuracy is returned as the solution. The main factors involved in the ACO are the setting up of search space, initialization of pheromone values, and generation of solutions, fitness evaluation of the generated solutions, pheromone evaporation and pheromone updating.

10.3 SEARCH SPACE FOR ACO IN PROPOSED ALGORITHM

The FSS, like any other problem, needs a corresponding ACO search space. Defining search space is one of the most important factors for getting better results from the algorithm. The algorithm is heavily dependent on the provided search space. The search space is according to the given dataset and is an N*N graph where N is the total number of attributes present in the dataset excluding target attribute. The nodes represent the features and the connection between the nodes i.e. edges, when traversed by the ant, denote the choice of next node, i.e. next feature. In this case, pheromone and heuristic value are not associated with links. Instead, each feature has its own pheromone value and heuristic value.

ISSN 2229-5518

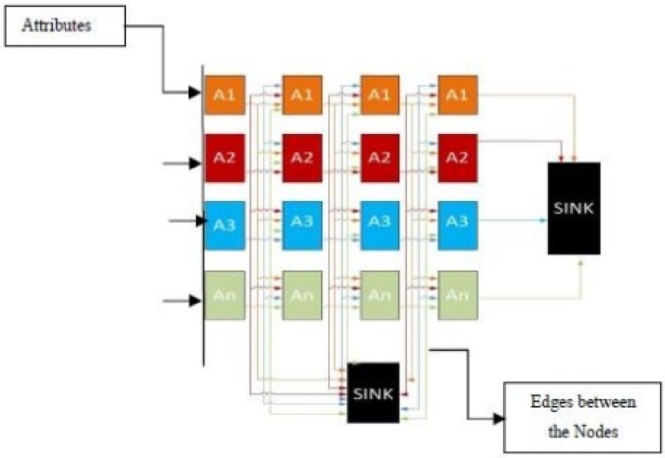


Figure: N*N Search Space for Ant.

11. INITIALIZATION OF PHEROMONE VALUES

The presence of pheromone values on the edges is the basic component of the ACO. Initially it is initialized by some small random value. In the experiments, the pheromone values on all edges are initialized at the start of the algorithm with the same amount of pheromone. In this way no attribute is preferred over other attributes by the first ant.

12. SELECTION OF AN ATTRIBUTE

The basic ingredient of any ACO algorithm is a constructive heuristic for probabilistically constructing solutions. A constructive heuristic assembles solutions as sequences of elements from the finite set of solution components. A solution construction starts with an empty partial solution. Then, at each construction step the current partial solution is extended by adding a feasible solution component from the set of solution components. A suitable heuristic desirability of traversing between features could be any subset evaluation function for example, an entropy based measure or rough set dependency measure. In this algorithm entropy based measure is used as heuristic information for feature selection. The heuristic desirability of traversal and node pheromone levels are combined to form the so-called probabilistic transition rule, denoting the probability that ant k will include feature i in its solution at time step t:

$$P_i^k(t) = \frac{[\tau_i]^\alpha . [\eta_i]^\beta}{\sum_{u \in j^k} [\tau_u]^\alpha . [\eta_u]^\beta} \quad \text{if } i \in j^k$$

where j^k is the set of feasible features that can be added to the partial solution; si and gi are respectively the pheromone value and heuristic desirability associated with feature i. a and b are two parameters that determine the relative importance of the pheromone value and heuristic information.

The transition probability used by ACO is a balance between pheromone intensity (i.e. history of previous successful moves), si, and heuristic information (expressing desirability of the move), gi. This effectively balances the exploitation–exploration trade-off. The search process favours actions that it has found in the past and which proved to be effective, thereby exploiting knowledge obtained about the search space. On the other hand, in order to discover such actions, the search has to investigate previously unseen actions, thereby exploring the search space. The best balance between exploitation and exploration is achieved through proper selection of the parameters a and b. If a = 0, no pheromone information is used, i.e. previous search experience is neglected. The search then degrades to a stochastic greedy search. If b = 0, the attractiveness (or potential benefit) of moves is neglected.

13 ASSUMPTIONS AND DEPENDENCIES

It is assumed that the user always types in his conscious.
It is assumed that all the hardware parts for feature extraction are flawless and work properly.
The accuracy of system also depends on the rate of change in features of a particular user.

14. IMPLEMENTATION DETAILS

14.1 PICTORIAL RESULTS

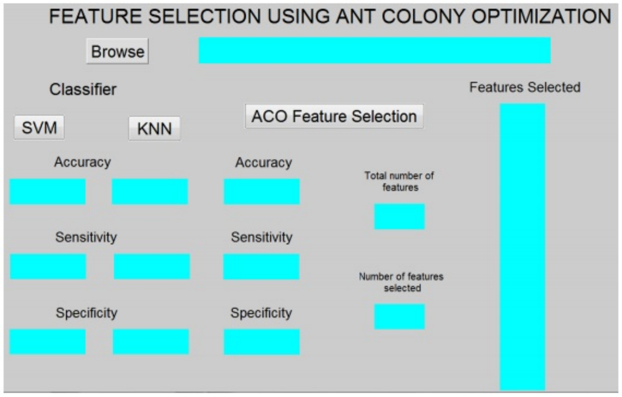


Figure: GUI of System

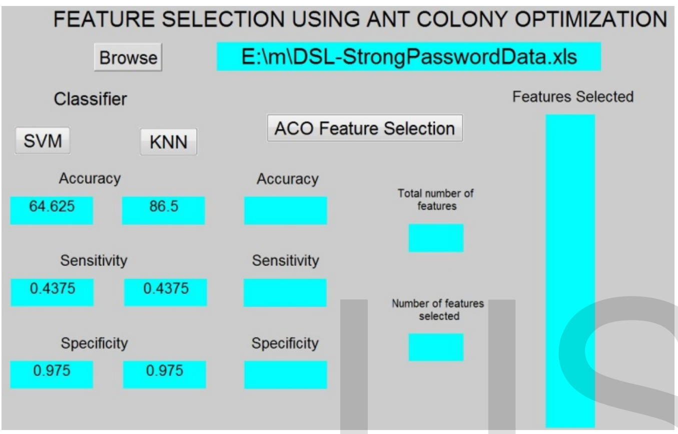


Figure: GUI of comparison of classifier

ISSN 2229-5518

15. RESULTS

	SVM	*KNN*
Accuracy	*64.625*	*86.5*
Sensitivity	*0.4375*	*0.4375*
Specificity	*0.975*	*0.975*

Table: Comparison of Classifier

	Before Feature Selection	*After Feature Selection(ACO)*
Accuracy	*64.625*	*64.875*
Sensitivity	*0.4375*	*0.4875*
Specificity	*0.975*	*0.968*

Table: Comparison before and after feature selection

16. CONCLUSION

16.1 PERFORMANCE EVALUATION

The two classifiers i.e. KNN and Multi-SVM have different outputs on the same input training dataset. Finally, after feature subset selection using ACO, the accuracy of the results have not only maintained but improved. After feature subset selection, the required time for authentication using subset of features would take very less time.

16.2 FUTURE DIRECTIONS

The classifiers and feature subset selection technique that has been used can also be used on different types of authentication techniques that has many features and classes. This would reduce the processing time and maintain the accuracy along.

17. ACKNOWLEDGEMENT

I owe special debt of gratitude to Mrs. Gunjan Pahuja, Department of Computer Science & Engineering, JSS Academy of Technical Education, Noida for her constant support and guidance throughout the course of work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for me. It is only her cognizant efforts that my endeavours have seen light of the day. I also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the study. Last but not the least, I acknowledge my friends for their contribution in the completion of the research work.

19. REFERENCES

[1] Dash M and Liu M , —Feature Selection for Classification,‖ in Department of Information System & Computer Science, National University of Singapore, Singapore 119260

[2] Forman G, —An Extensive Empirical Study of Feature Selection Metrics for Text Classification,‖, Journal of Machine Learning Research in Hewlett-Packard Labs Palo Alto, CA, USA 94304

[3] Kohavi R,, George H. John b, —Wrappers for feature subset selection ,‖ in Data Mining and Visualization, Silicon Graphics, Inc., 2011 N. Shoreline Boulevard, Mountain view, CA 94043, USA

[4] Karnan, M. and Akila, M., "Personal Authentication Based on Keystroke Dynamics Using Soft Computing Techniques",Pp.334-338, 2010.

[5] D. Shanmugapriya, and G. Padmavathi , —An Efficient Feature Selection Technique for User Authentication using Keystroke Dynamics,‖ in IJCSNS International Journal of Computer Science and Network Security, Vol.11 No.10, October 2011

[6] Enzhe Yu and Sungzoon Cho, —Keystroke Dynamics Identity Verification - Its Problems and Practical Solutions‖, Vol. 23, 2004.

[7] Gabriel L. F. B. G. Azevedo, George D. C. Cavalcanti and E.C.B. Carvalho Filho, —An Approach to Feature Extraction for Keystroke Dynamics Systems based on PSO and Feature Weighting‖, Pp. 3577-3584, 2007.

[8] Mehdi Hosseinzadeh Aghdam, Nasser Ghasem-Aghaee, Mohammad Ehsan Basiri,"Text feature selection using ant colony optimization—, Vol. 36, 2009

[9] İbrahim Sogukpinar Levent Yalcin , — User Identification at log on via Keystroke Dynamics‖

[10] L. Yiu and F. Z. Yuan, —One-Against-All Multi-Class SVM Classification Using Reliability Measures:‖ at The Ohio State University

[11] P´adraig Cunningham and Sarah Jane Delany, —k-Nearest Neighbour Classifiers,‖ in University College Dublin

[12] Kevin S. Killourhy and Roy A. Maxion. "Comparing Anomaly Detectors for Keystroke Dynamics," in Proceedings of the 39th Annual International Conference on Dependable Systems and Networks (DSN-2009), pages 125-134, June 2009

[13] http://in.mathworks.com/discovery/feature-selection.html

[14] http://www.cs.cmu.edu/~keystroke/

[15] http://in.mathworks.com/help/stats/support-vector-machines-svm.html

[16] http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

INDEX

i. Introduction
ii. Objective
iii. Scope
iv. Keystroke Dynamics Analysis Framework
v. Features in Keystroke Dynamics
    i. Dwell Time(DT) or Duration
    ii. Flight Time(FT)

ISSN 2229-5518

- *Shivam Sharma has completed his B.Tech, Computer Science and Engineering from JSS Academy of Technology, Noida PH-08010206467.*
  E-mail: sharmashivam2806@gmail.com