SecureNet: A PySpark-Based Approach to Enhanced Network Intrusion Detection Using Machine Learning and Feature Engineering

B. Akhileswar Reddy, G. Someswara Reddy, K. Lokesh, Manju Venugopalan\*

*Department of Computer Science and Engineering,  
Amrita School of Computing, Bengaluru,  
Amrita Vishwa Vidyapeetham, India*

[*akhileswarreddy5016@gmail.com*](mailto:akhileswarreddy5016@gmail.com) *,* [*somu.gopireddy@gmail.com*](mailto:somu.gopireddy@gmail.com) *,* [*kolleparalokesh@gmail.com*](mailto:kolleparalokesh@gmail.com)*,* [*v\_manju@blr.amrita.edu*](mailto:v_manju@blr.amrita.edu)

**Abstract:** **Cyber-attacks are increasing day to day in this world of AI and Internet.** **A ML model to detect the network attack based on the network parameters is of great use.** **The proposed model uses a new and complete method to make intrusion detection in computer networks more accurate. Using PySpark's shared computing power, machine learning rules are put in the network data. It helps to make a strong first step of sorting things out. To optimize and streamline the feature space, two distinct techniques are employed sequentially; Chi-Square Selector for selecting important features and Principal Component Analysis to reduce features. These methods are used one after the other to improve how well machine learning models work. This makes them more accurate and flexible for finding unauthorized attacks on computer systems. It adds important value to the world of cybersecurity and Big Data analysis. From the experiments. The model which reported the best performance is Multi-Layer Perceptron with 0.993 accuracy.**

Key words: Text classification, Feature Selection, Feature Reduction, Principle Component Analysis, chi square selector, PySpark, Machine learning models, SMOTE

# INTRODUCTION

Cybersecurity for networks [1,2,3,4] is a growing worry in our connected world today. Computer network safety keeps being threatened by bad activities from different sources. NID or Network Intrusion Detection is a very important defense system. It aims to find and stop unauthorized people from entering networks, as well as spot any strange behaviour in the network traffic. In the world of Big Data Analytics (BDA), network data has become much bigger and more confusing. This means we need new ways to quickly find out if someone is breaking into our system during real-time, using strong methods. This project works on using ML methods in PySpark to solve the problem of studying big labelled network data for correctly placing situations as normal or strange. This task wants to use PySpark's spread out computer skills in order to manage huge sets of data with labels correctly. This project uses ML techniques like Random Forest, Logistic Regression and Gradient Boosting to look at network traffic details. It wants to make strong models that can tell the difference between regular network behaviour and odd patterns. Furthermore, the addition of feature engineering methods made for PySpark aims to make models better. This ensures that network intrusions are caught correctly. This research uses PySpark-based machine learning methods to handle large amounts of labelled data. It aims for a better and faster Network Intrusion Detection system under the scope of Big Data Analytics.

At the start of our project, we used many types of machine learning tools on our data set. This helped us understand how well basic class sorts worked without any extra help or improvements. Random Forest, Logistic Regression and Gradient Boosting are used to look at network traffic data. These algorithms help classify different situations. We saw that the network data is complicated and multi-layered. So, we added methods to reduce features in order to make our model work better and faster. PCA was used as a strong way to reduce features in the PySpark place [5,6,7,8]. This helped us keep only important parts of data while also saving on workload time. Starting with the first results from machine learning, we began a step-by-step process. This included simplifying features and checking our models again. The data set, not one but two times had features removed. This was done using a method called PCA on both occasions. This step-by-step way helped us find the best balance between how good our model is and its speed.

We know how important picking the right features is for understanding a model and making it work better. So, we used something called Chi-Square selector in our plan to do just that. This maths way helped us to find and keep the best elements for stopping bad attacks, making machine learning models more focused. We used the PySpark platform for big computers to deal with large-scale and complicated network data sets. The power of PySpark helped with fast, easy teamwork and learning machines. It made sure that big data could be analysed quickly from lots of information labelled together in large analysis setups. This research is a complete study of how machine learning (ML) can help find security problems. It combines different ways, uses strategies to remove unnecessary features and chooses the best ones all within PySpark program frame. The next parts go into the ways, outcomes and talks that come from every step. They help us understand how our method is possible or works well.

The key contributions of the paper are:

1) Using PySpark for distributed data processing and analysis to efficiently handle large-scale datasets, perform complex computations, and leverage parallel processing capabilities.

2) Random Forest, Decision Tree, SVM, Naive Bayes, and MLP were employed in the project for diverse tasks including ensemble learning, hyperplane construction, probabilistic classification, and deep learning for classification.

# II. RELATED WORKS

The current section reports the findings of a survey confined to network intrusion detection models which are ML and DL based.

A prominent work [9] explores Support Vector Machines (SVM) as a classifier and adjusts the settings. The main goal is to make network security stronger by improving the choice of features, designing classifiers better and adjusting parameters for improved results. This happens with single objective optimization methods like finding useful feature sets or detailed designs like filter or wrapper options. The research wants to solve problems with detecting network attacks by using a complete method that includes making models better and picking important features. Liu and team have explored on how to spot and group online attacks using a method [10] called Support Vector Machines (SVM) and another one called Principal Component Analysis. This is based on the actions of network traffic behaviour. The main goal is to rightly sort out bad items and have very few false alarms. They made a model in an easier way for Network Intrusion Detection Systems (NIDS) using just two groups or classifications, showing how good their idea was through tests on the same records set. They study and test key classification features in detail, emphasizing the importance of using ML tools for security breach detection. Kiran and Ajmeer was explained [11] about a smart system built on knowledge looks like, focusing mainly on the rule resolving part's important role in spotting unauthorized access. The use of smart ways, like expert systems, is shown. A key point talks about how the rule reading part works with data. The paper also looks at using deep learning models like Convolutional Neural Networks (CNN) to improve the accuracy and flexibility in auditing. The main goal is to make security better by using smart ways of describing hacking events and advanced learning models. It suggests a better LSTM neural network classification model [12]. This uses features chosen by QPSO to lower the number of details and make a thick feature list. The main goal of the study is to make it better and faster at spotting network attacks. It does this by handling difficult features in complex internet traffic patterns. This way is better than other ways to find things, backed up by higher F1 scores and lower false good rates when checked on real Internet traffic data. QPSO helps chose important features and LSTM does the classifying. This pair makes it faster and more accurate to spot different threats that have unique timing patterns.

The paper is about using machine learning models [13] to find new and zero-day cyber attacks. It suggests a mix method using Simple Bayes, Support Vector Machine and Random Forest classifiers to better find out attacks. It also helps stop inside threats from happening. The main goal is to improve checking for intrusions by studying user forensic details and security measures, especially using the CRF and spider monkey optimization methods. They help choose which features are most important in spotting sneaky hacking attempts. The design has two parts for admin and user use. SMO is used to decide the important features, while CNN helps tell apart attacks from normal activity. This article talks about [14] a study that works on making an intrusion detection system really good. The main goal of the research is to find and stop network attacks efficiently. To do this, the study uses Random Forest as its main method for finding unwanted access. The model uses parts of network data and the traits of intrusion episodes to correctly spot possible dangers for Networks. The study stresses the need to find and stop wrong people from getting into networks. A Random Forest method is used as a strong way to deal with this big security problem. Galatro and team has explored different ways to pick out features for intrusion detection [15]. Its main goal is to examine various FS methods in detail. The paper gives a complete view of network traffic analysis and safety methods. It compares popular FS algorithms on speed, connections between features and how long they take to work out results that can help with overseeing networks/safety control. A prominent work [16] explores how to improve NIDS works by using different learning techniques from machines through SDN. The study thoroughly checks how well-known and effective machine learning methods (RF, SVM, Decision Tree, and Naive Bayes) work with the security invasion detection data set. It includes steps like data counting, cleaning up, making features the same size and choosing them to make sure learning goes well. The goal is to get the best results with less data representation and size. This study helps choose great algorithms that match different types, sizes of data, and network behavior well.

The part from the paper talks about different ways [17] to spot network break-ins. This includes studying attack plans on traffic rules and using moving information patterns. It also explains the idea of weighted average for mistakes in classifying things. It points out how punishing wrong alarms and detecting rates are different if all weights are equal. Furthermore, it points to a study on web team learning and mentions an exact Ada-boost method for network intrusion detection. The paper talks about using advanced learning methods, like convolutional neural networks (CNN) and quick-learning without much data (FSL), for systems that detect bad things happening on computer networks. It shows the results of tests proving these methods have great accuracy and detail in identifying hacker attacks. Moreover, the paper shows how to improve protection for networks and reduce security worries in today's life. Even though the results are impressive, there's a warning about how easy it is for bad actors to attack systems that use deep learning in network management. The new models being suggested are much better than old ways for detecting attacks. They also show what could be done in the future to make this method faster and more accurate by using transfer learning. Ahamad and Zeeshan has explored studies on systems that watch for attacks in computer networks using machine learning and deep learning methods. Different studies are shown. Each one uses different methods and ideas, such as recurrent neural networks, group methods improved with BAT algorithm optimization, non-symmetric deep autoencoder paired with random forest learning style and fast network study followed by particle swarm idea. More types studied are based on data mining algorithms called 'deep neural nets'. The good and bad points of these ways are also looked at, showing how they can get caught quicker or be trained faster but may not work as well with different kinds of attacks [18]. The article is about making a system to watch for computer security problems using machine learning, with special focus on choosing good features quickly. It talks about getting data ready, using the Chi squared Test method to choose features and building Intrusion Detection Systems on Advanced Security Network Metrics database with Logistic Regression and Neural Networks. The special thing about this paper is showing how you can use ways to pick important features. This reduces the model's complexity and speeds up training time, trying to close a gap in writing from earlier times on using machine learning methods for datasets where extra information has been made and these kinds of feature selection purposes are used too.

The survey reviews ML and DL-based network intrusion detection models, highlighting techniques such as Support Vector Machines (SVM), Principal Component Analysis, Convolutional Neural Networks (CNN), and Random Forest. It emphasizes feature selection methods, optimization algorithms like QPSO and BAT, and the use of advanced learning models to enhance accuracy and efficiency in detecting cyber-attacks. The findings underscore the importance of improving network security through smart feature selection, model optimization, and the adoption of deep learning techniques, despite challenges such as susceptibility to attacks on systems utilizing deep learning. The following sections explains proposed methodology which experiments ML models in combination with feature engineering techniques like feature selection(chi-square) and feature reduction (PCA) and applying classifiers.

# METHODOLOGY

The proposed methodology is explained in detail in the following subsections and showcased in Fig.1.

## *A. Dataset Description*

The dataset is sourced from Kaggle[[1]](#footnote-1) which is of 25,000 rows and 42 columns. It provided an environment for obtaining raw TCP/IP dump data for a network by emulating a typical US Air Force LAN. The LAN was focused like a real setting and bombarded with many attacks. A connection is a sequence of TCP packets that begin and finish at a specific time interval, during which data travels to and from a source IP address to a target IP address using a well-defined protocol. Furthermore, each link is classified as either normal or an attack, with just one specific attack type. Each connection record consists of around 100 bytes.

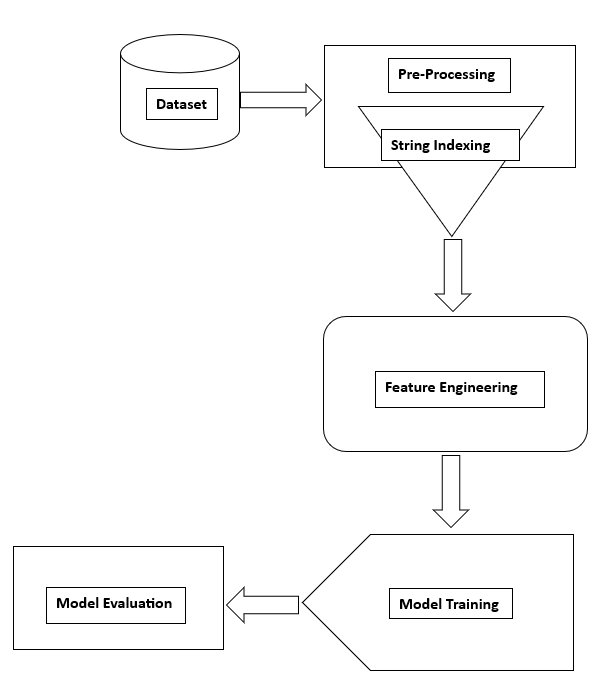


Fig.1 Proposed Methodology of Network Intrusion Detection.

## *B. Pre-Processing*

The pre-processing stage included string indexing which converted categorical features in the dataset like *protocol\_type, service, flag, class* into numerical indices. This step was crucial for algorithms that require numerical input, as it transformed categorical labels into a format suitable for machine learning models.

## *C. Feature Engineering*

### *a. Feature Selection using Chi-Square Selector*

Chi-Square based feature selection is applied to select the top 25 features based on the label. It’s a powerful tool in the machine learning toolbox, which works especially well for feature selection in problems where the variables are categorical. It evaluates each feature's relationship to the target variable by utilizing the chi-square test, a statistical measure of independence. The selector aids in the removal of superfluous or irrelevant features, hence improving the interpretability and performance of the model, by selecting features that show strong relationships with the target.

### *b. PCA (Principal Component Analysis)*

PCA is implemented to perform dimensionality reduction on the standardized features. Created a pipeline that included feature assembling, scaling, and PCA. Fitted the model on the data and transformed the dataset to include PCA features.

## *D. Model Training*

Divided the dataset into training and testing sets in the ratio 70:30. The different classifiers experimented are trained on the training data.

## *E. Classification*

**Naive Bayes:** Naïve Bayes is a predicting way of sorting things based on rules from math with the simple idea that each feature stands alone. It is an odds-based program that's often used for sorting text, stopping spam and other work where we can show data with chances.

**Support Vector Machines:** SVM are a type of supervised learning method used for sorting and calculating numbers. SVMs work really well in spaces with many dimensions and are very popular for machine learning and figuring out patterns. The main idea behind SVM is to find a line that best separates data into different groups.

**Decision Tree:** A Decision Tree is a supervised learning method used for two types of tasks, where it classifies objects or predicts numbers. It works by dividing the data into smaller groups again and again. At each step, it looks at which feature is most important first. The aim is to make a tree-like model that makes choices by going down the branches of the tree.

**Random Forest:** Random Forest is a group learning way that makes many decision trees during training. It then picks the most common result (classification) or average prediction (regression) from these individual trees to give its answer. It is one of the most liked and strong machine learning ideas because it can adapt well and is tough.

**Multi-Layer Perceptron:** MLP is a kind of fake brain network structure used in computer learning to make machines smarter. MLPs are a type of feedforward neural network. This means that information only moves in one way - from the input part, through hidden layers and then to the output part.

## *F. Model Evaluation*

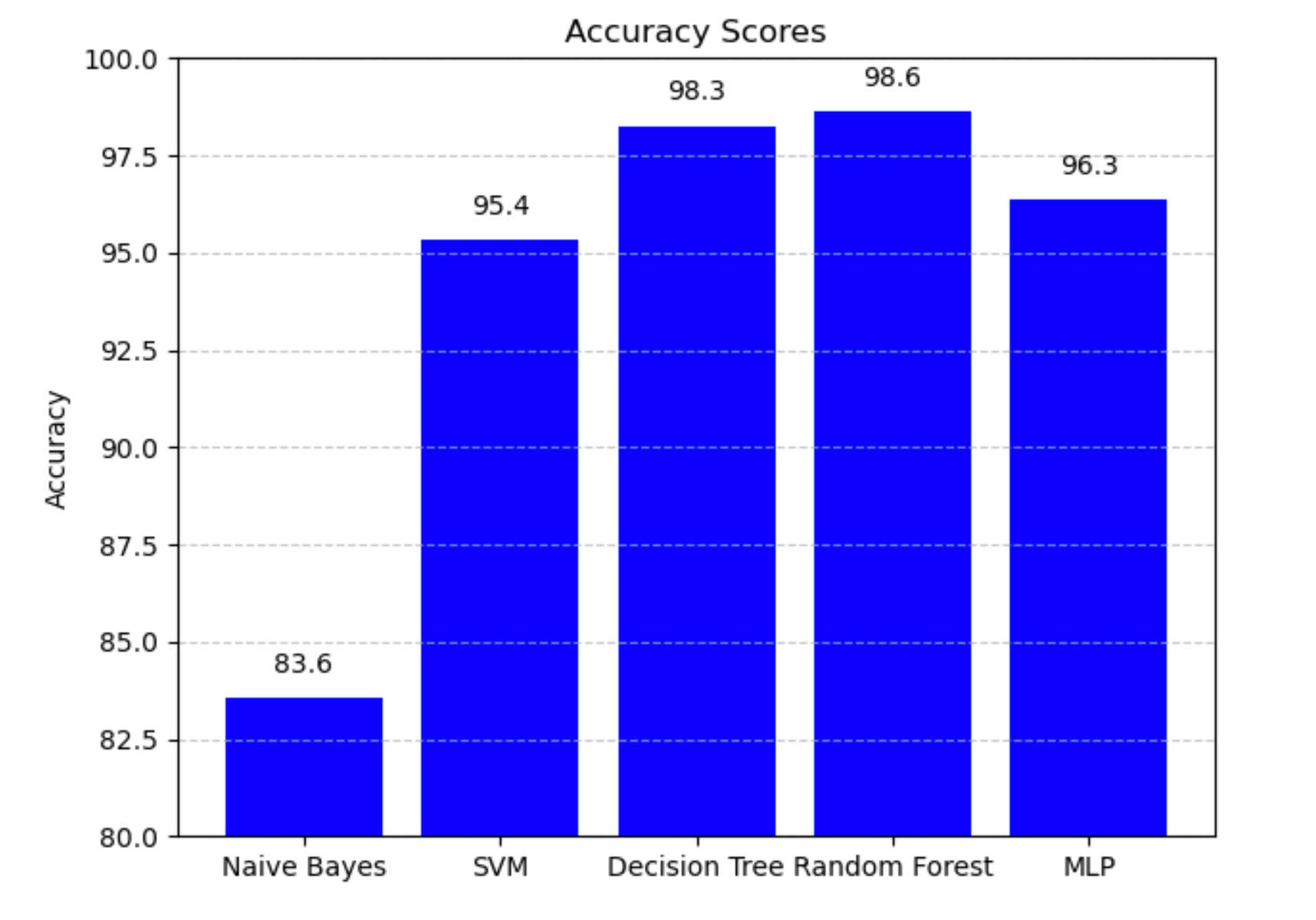
The performance of each algorithm on the testing data is evaluated using different classification metrics such as accuracy, precision, recall, and F1-score. These metrics provided insights into how well the model performed in terms of correctly classifying instances.

# IV. RESULTS AND ANALYSIS

The experiments performed included assessing the contribution of every phase performed. Initially, the trained classifiers are applied on the pre-processed data, the results of which are displayed in Table 1. Random Forest Classifier appeared to be the best with F1 score of 98.47% as shown in Table 1. Decision Tree has is almost in par with F1 score of 98.26%. The accuracies and F-measures reported by the experimented models are visually plotted in Fig.2 and Fig.3 respectively.

Table 1: Performance Metrics of Classifiers applied on the Pre-Processed Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1** |
| Naive Bayes | 83.56 | 84.52 | 83.56 | 83.56 |
| SVM | 95.35 | 95.36 | 95.35 | 95.34 |
| Decision Tree | 98.26 | 98.26 | 98.26 | 98.26 |
| **Random Forest** | **98.47** | **98.51** | **98.47** | **98.47** |
| MLP | 96.34 | 96.37 | 96.34 | 96.34 |

Fig.2: Graphical Representation of Accuracy scores of the models applied on Pre-Processed Data

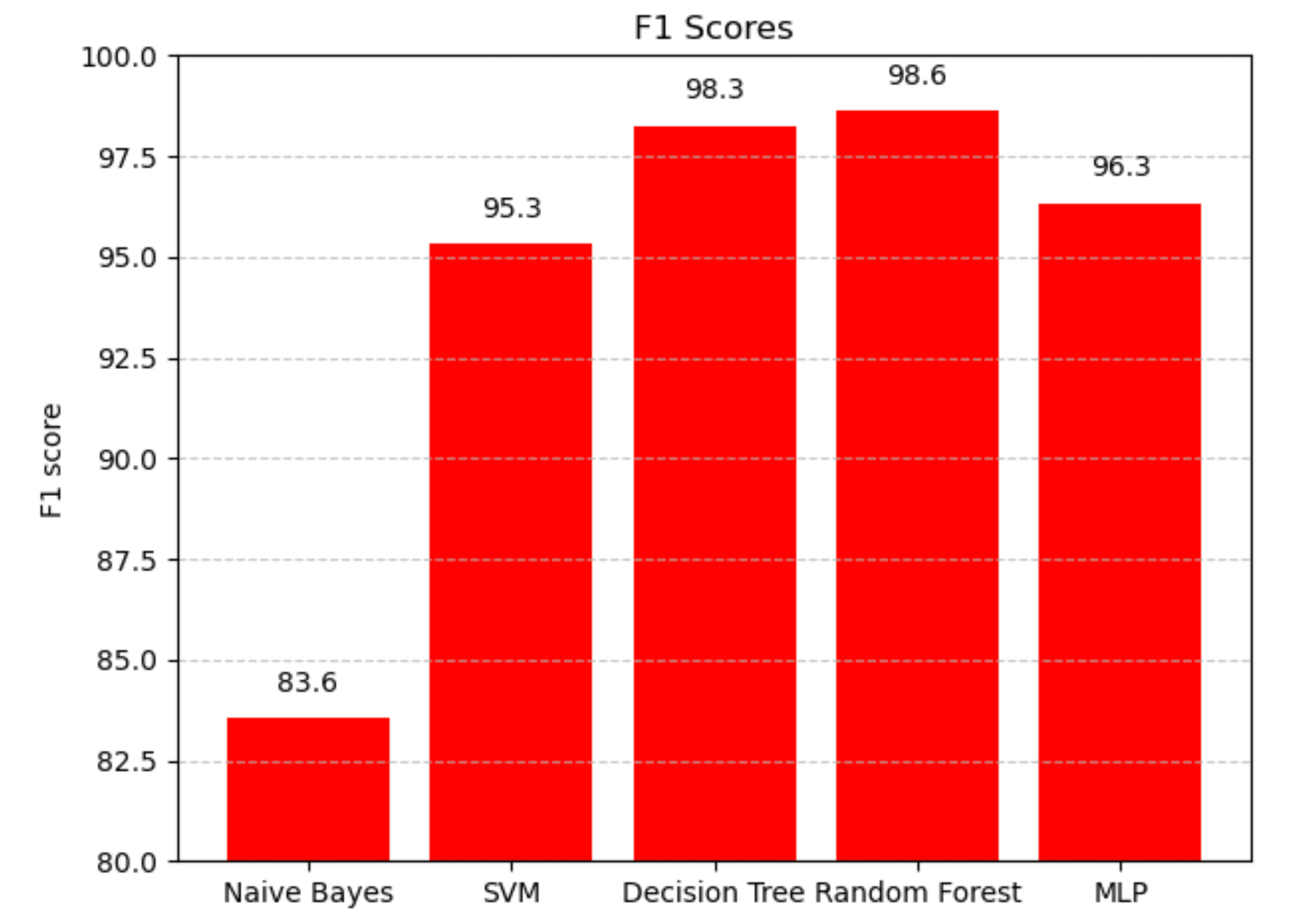


Fig.3: Graphical Representation of F1 Scores of the models applied on Pre-Processed Data

For the next set of experiments, the classifiers are trained on applying feature selection (Chi-Square) by selecting top 25 features, the results of which are displayed in Table.2. MLP and Decision Tree got the best results with F1-score of 99.24%. By applying feature selection results have improved when compared to the results in Table.1. The accuracies and F-measures reported by the experimented models are visually plotted in Fig.4 and Fig.5 respectively.

Table 2: Performance Metrics of Classifiers after the applying feature selection (Chi Square Selector)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1** |
| Navie Bayes | 65.25 | 72.00 | 65.25 | 60.58 |
| SVM | 94.43 | 94.43 | 94.43 | 94.43 |
| Decision Tree | 97.70 | 97.75 | 97.70 | 97.71 |
| **Random Forest** | **98.43** | **98.45** | **98.43** | **98.44** |
| MLP | 96.08 | 96.14 | 96.08 | 96.08 |

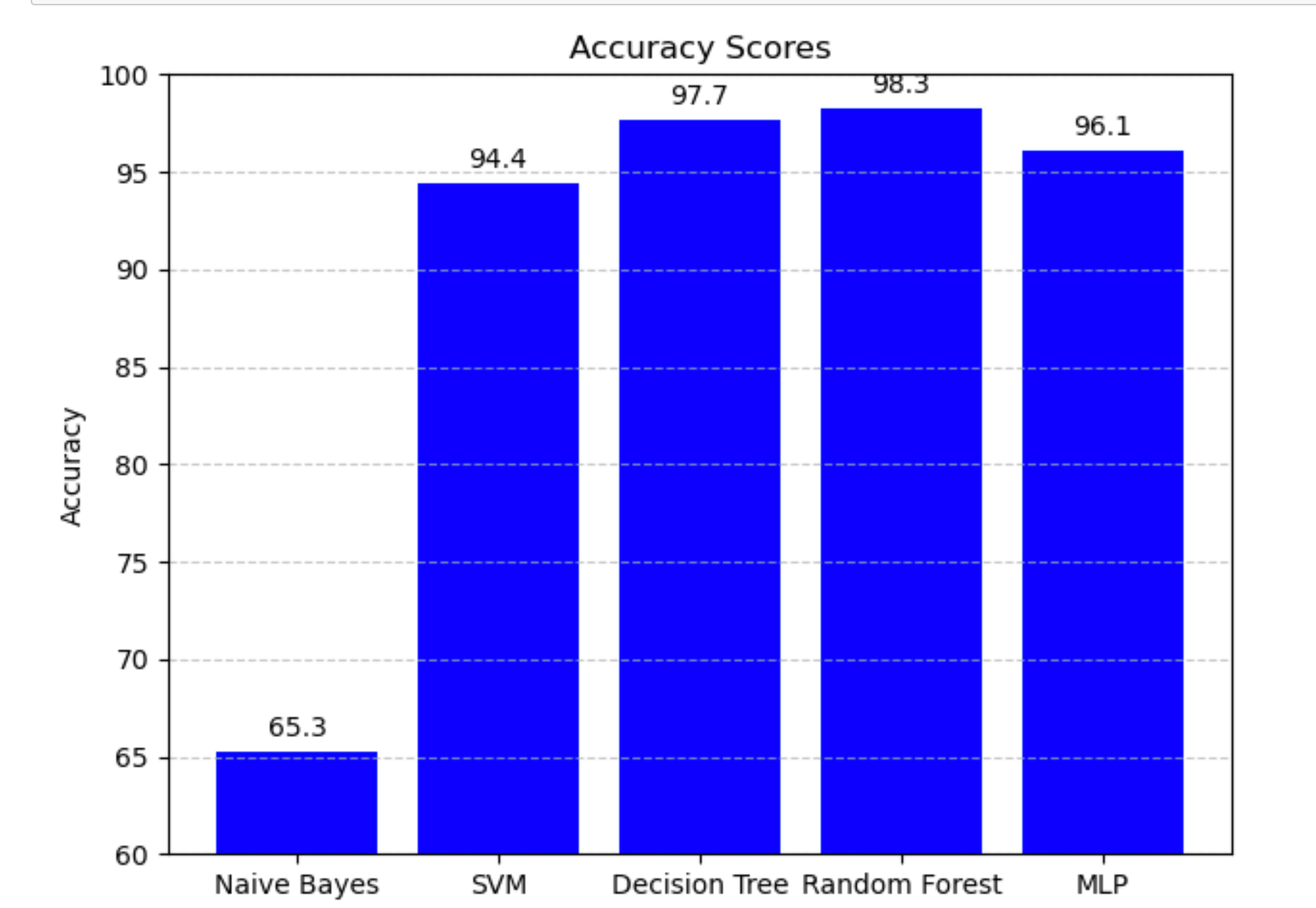


Fig.4: Graphical Representation of Accuracy Scores of the models after applying Feature Selection (Chi Square Selector)

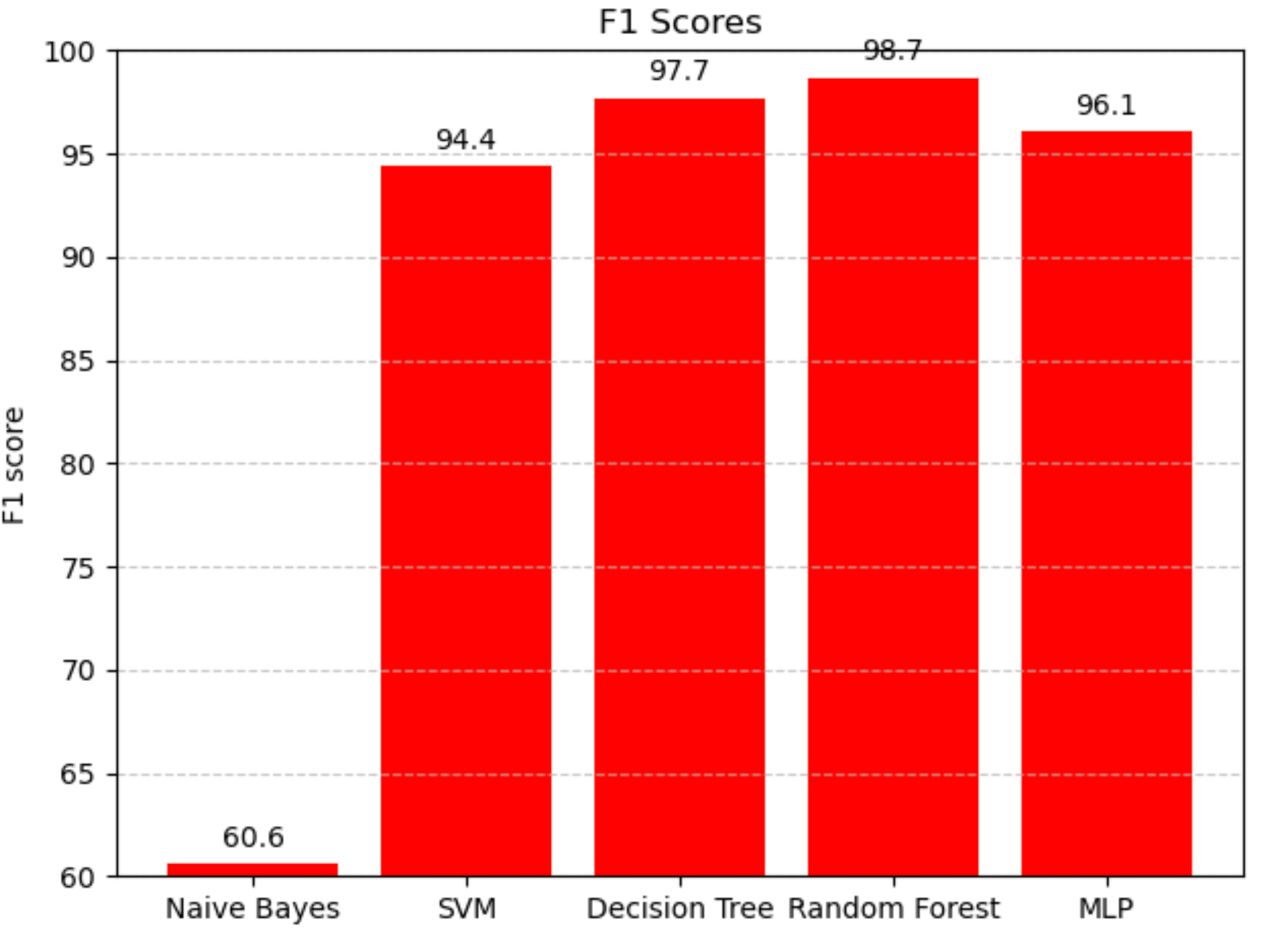
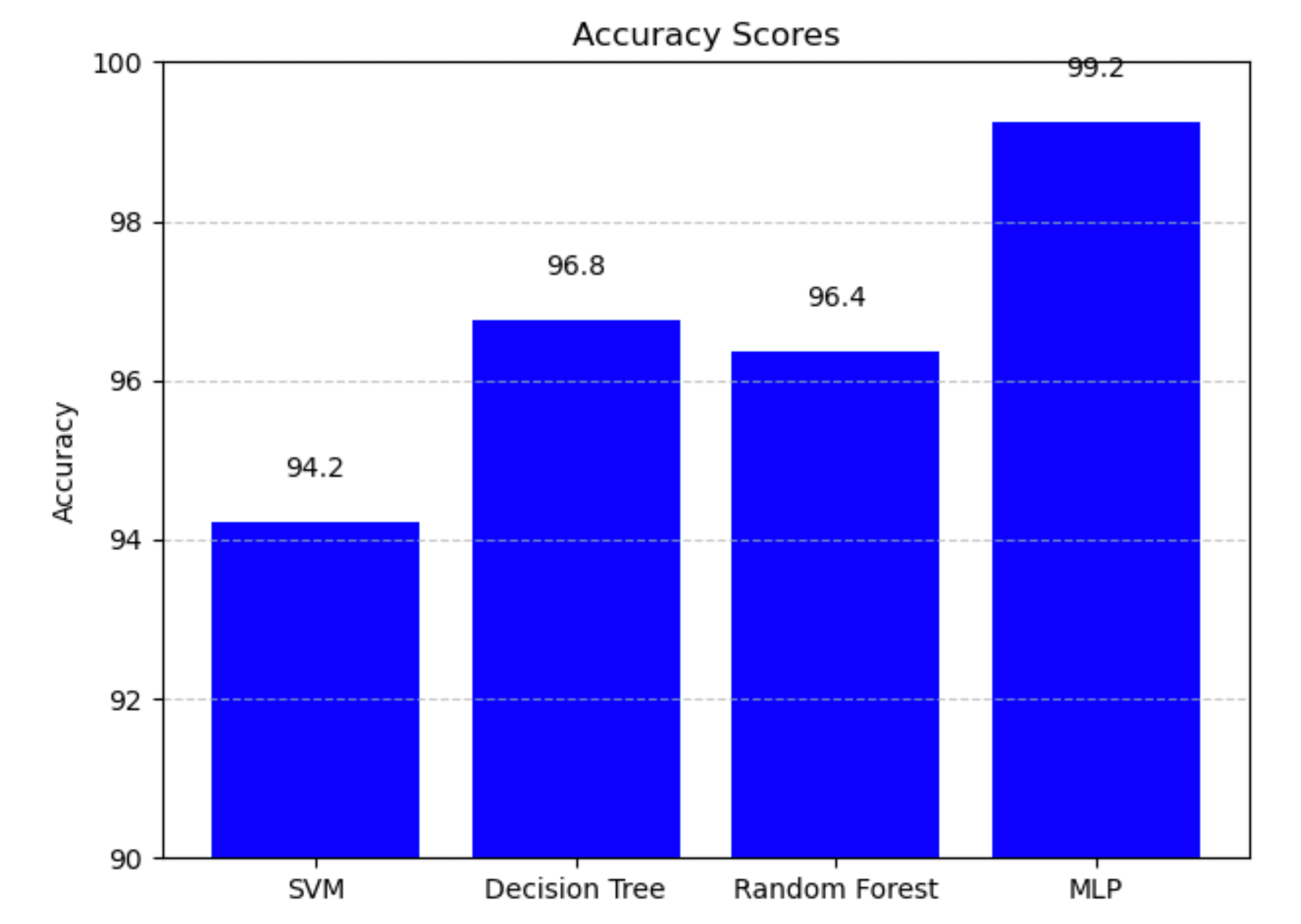


Fig.5: Graphical Representation of F1 Scores of the models after applying Feature Selection (Chi Square Selector)

The next set of experiments included applying Feature Reduction (PCA) for dimensionality on pre-processed data, the results which are displayed in Table 3. Random Forest achieved the best results with F1-score of 98.44%. Naive Bayes can’t be applied for the data after applying PCA as it gives Negative values. Naïve bayes is based on probabilistic Functions which doesn’t support Negative values. The accuracies and F-measures reported by the experimented models are visually plotted in Fig.6 and Fig.7 respectively.

Table 3: Performance Metrics of Classifiers after the applying Feature Reduction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| SVM | 94.35 | 94.35 | 94.35 | 94.35 |
| Decision Tree | 95.73 | 95.73 | 95.73 | 95.73 |
| Random Forest | 96.28 | 96.47 | 96.28 | 96.27 |
| MLP | **99.30** | **99.30** | **99.30** | **99.30** |

Figure 6: Graphical Representation of Accuracy Scores of the models after applying Feature Reduction

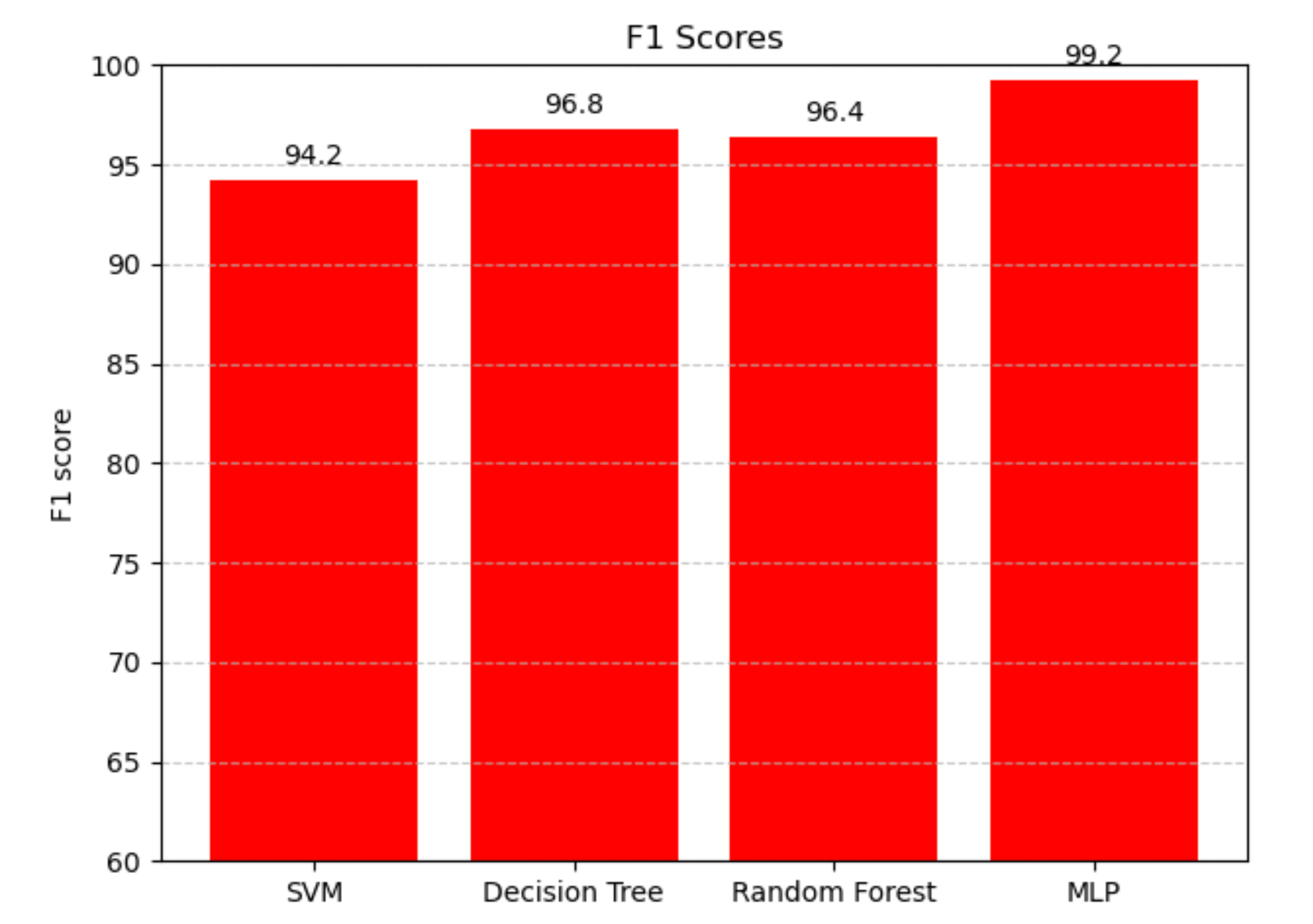


Fig.7: Graphical Representation of F1 Scores of the models after applying Feature Reduction(PCA)

The next set of experiments included applying Feature Reduction (PCA) for dimensionality followed by feature selection, the results which are displayed in Table 4. MLP got the best result with F1-score of 99.30%. T The accuracies and F-measures reported by the experimented models are visually plotted in Fig.8 and Fig.9 respectively. The proposed model performed best after applying Chi-square selection followed by PCA based dimensionality reduction which can be projected as the proposed model.

Table 4: Performance Metrics of Classifiers after the Feature Selection Followed by Feature Reduction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1** |
| SVM | 94.21 | 94.21 | 94.21 | 94.21 |
| Decision Tree | 99.24 | 99.24 | 99.24 | 99.24 |
| Random Forest | 96.58 | 96.68 | 96.58 | 96.57 |
| **MLP** | **99.24** | **99.24** | **99.24** | **99.24** |

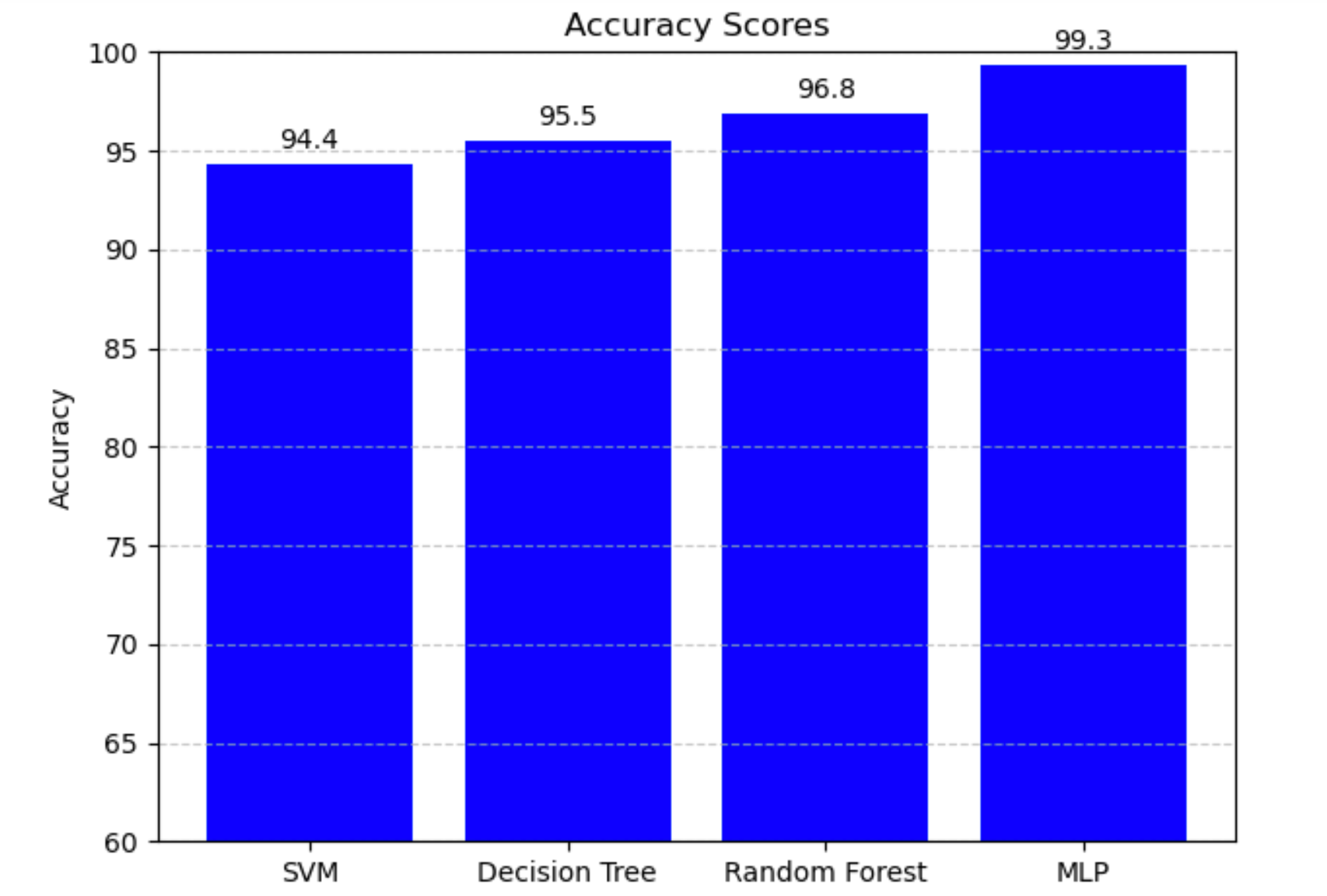


Fig.8: Graphical Representation of Accuracy Scores of the models after the Feature Selection Followed by Feature Reduction

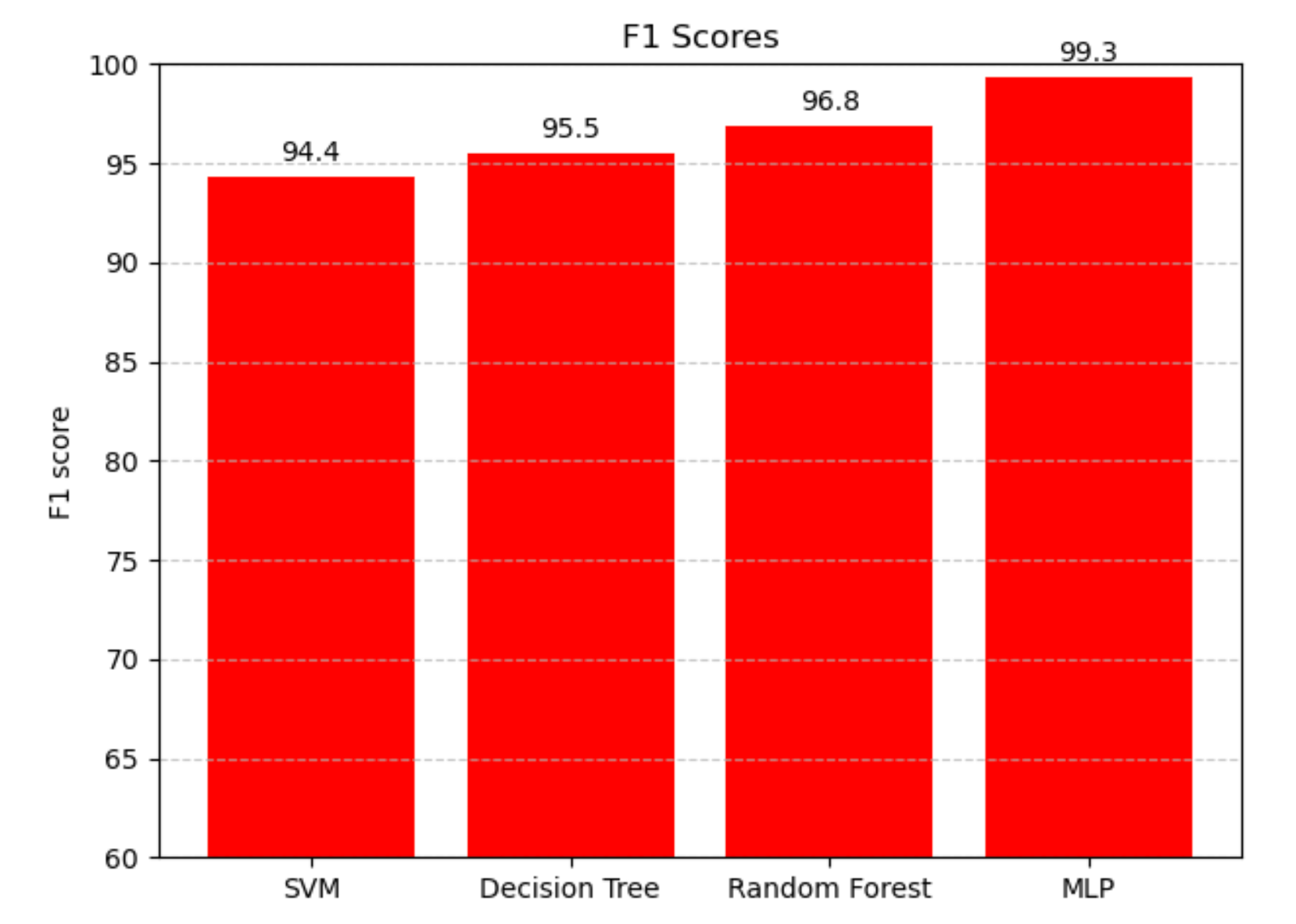


Fig.9: Graphical Representation of F1 Scores of the models after the Feature Selection Followed by Feature Reduction

##### V. CONCLUSION AND FUTURE WORK

The proposed model experimented multiple ML models and explored feature selection and feature reduction methods and hence identified the best model as Multi-Layer Perceptron which reported an F-measure and accuracy of 99.3% after incorporating Chi-Square based feature selection and PCA based feature reduction.

The proposed model can be extended by training with Deep Learning models and ensemble models and also the proposed model can be deployed in a real-world network environments and can conduct extensive evaluation and validation of the intrusion detection system.

##### REFERENCES

1. Hu W, Hu W, Maybank S. AdaBoost-based algorithm for network intrusion detection. IEEE Trans Syst Man Cybern B Cybern. 2008 Apr;38(2):577-83. doi: 10.1109/TSMCB.2007.914695. PMID: 18348941.
2. [2] Dongdong, Liu, Dou Hongtao, Han Bo, and Niu Lei. "An optimized network intrusion detection model." In 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), vol. 5, pp. 227-232. IEEE, 2022.
3. [3] Kiran, Ajmeera & Prakash, S. & Kumar, B & Likhitha, & Sameeratmaja, Tammana & Charan, Ungarala. (2023). Intrusion Detection System Using Machine Learning. 1-4. 10.1109/ICCCI56745.2023.10128363.
4. [4] V. Hnamte and J. Hussain, "Network Intrusion Detection using Deep Convolution Neural Network," 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-6, doi: 10.1109/INCET57972.2023.10170202.
5. [5] V. Sidharth and C. R. Kavitha, "Network Intrusion Detection System Using Stacking and Boosting Ensemble Methods," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 357-363, doi: 10.1109/ICIRCA51532.2021.9545022.
6. [6] T. N. Varunram, M. B. Shivaprasad, K. H. Aishwarya, A. Balraj, S. V. Savish and S. Ullas, "Analysis of Different Dimensionality Reduction Techniques and Machine Learning Algorithms for an Intrusion Detection System," 2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA), Arad, Romania, 2021, pp. 237-242, doi: 10.1109/ICCCA52192.2021.9666265.
7. [7] B. Ganesh and S. Sridevi, "Analysis of Hybrid Deep Learning Models for Efficient Intrusion Detection," 2023 International Conference on Networking and Communications (ICNWC), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICNWC57852.2023.10127270.
8. [8] G. B. T. and S. Chokkalingam, "Enhancing Intrusion Detection in Wireless Sensor Networks Using Deep Learning-Based K-Barriers Prediction," 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2023, pp. 1208-1215, doi: 10.1109/ICSSAS57918.2023.10331818.
9. [9] B. S. Babu, G. A. Reddy, D. K. Goud, K. Naveen and K. S. T. Reddy, "Network Intrusion Detection using Machine Learning Algorithms," 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), Trichy, India, 2023, pp. 367-371, doi: 10.1109/ICSMDI57622.2023.00071.
10. [10] L. Zhang, H. Yan and Q. Zhu, "An Improved LSTM Network Intrusion Detection Method," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 2020, pp. 1765-1769, doi: 10.1109/ICCC51575.2020.9344911.
11. [11] M. Di Mauro, G. Galatro, G. Fortino, A. Liotta,Supervised feature selection techniques in network intrusion detection: A critical review,Engineering Applications of Artificial Intelligence,Volume 101,2021,104216,ISSN09521976,https://doi.org/10.1016/j.engappai.2021.104216.
12. [12] Alzahrani, Abdulsalam O., and Mohammed J. F. Alenazi. 2021. "Designing a Network Intrusion Detection System Based on Machine Learning for Software Defined Networks" Future Internet 13, no. 5: 111. <https://doi.org/10.3390/fi13050111>
13. [13] G. ZHU, H. YUAN, Y. ZHUANG, Y. GUO, X. ZHANG and S. QIU, "Research on network intrusion detection method of power system based on random forest algorithm," 2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Beihai, China, 2021, pp. 374-379, doi: 10.1109/ICMTMA52658.2021.00087.
14. [14] C. Lu, "Research on the technical application of artificial intelligence in network intrusion detection system," 2022 International Conference on Electronics and Devices, Computational Science (ICEDCS), Marseille, France, 2022, pp. 109-112, doi: 10.1109/ICEDCS57360.2022.00031.
15. [15] Ahmad, Zeeshan & Shahid Khan, Adnan & Shiang, Cheah & Ahmad, Farhan. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies. 32. 10.1002/ett.4150.
16. [16] R. Desai and V. T. Gopalakrishnan, "Network Intrusion Detection Through Machine Learning With Efficient Feature Selection," 2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS), Bangalore, India, 2023, pp. 797-801, doi: 10.1109/COMSNETS56262.2023.10041315.
17. [17] D. V C, N. K. S, H. V. V, S. K. T, G. K. T and S. Vajipayajula, "Comparative Analysis of Deep Learning and Machine Learning models for Network Intrusion Detection," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-13, doi: 10.1109/ICCCNT56998.2023.10308108.
18. [18] R. K. Vigneswaran, R. Vinayakumar, K. P. Soman and P. Poornachandran, "Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 2018, pp. 1-6, doi: 10.1109/ICCCNT.2018.8494096.

1. <https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection> [↑](#footnote-ref-1)