International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT)
Vol. 6, No. 1, March 2015

INTRUSION DETECTION USING INCREMENTAL LEARNING FROM STREAMING IMBALANCED DATA

Dipali Bhosale[1]and Roshani Ade[2]
[1]Dept of Computer Engg, DYPSOET, SavitribaiPhule Pune University, Maharashtra
[2]Dept of Computer Engg, DYPSOET, SavitribaiPhule Pune University, Maharashtra

ABSTRACT

Most of the network habitats retain on facing an ever increasing number of security threats. In early times, firewalls are used as a security examines point in the network environment. Recently the use of Intrusion Detection System (IDS) has greatly increased due to its more constructive and robust working than firewall. An IDS refers to the process of constantly observing the incoming and outgoing traffic of a network in order to diagnose suspicious behavior. In real scenario most of the environments are dynamic in nature, which leads to the problem of concept drift, is perturbed with learning from data whose statistical attribute change over time. Concept drift is impenetrable if the dataset is class-imbalanced. In this review paper, study of IDS along with different approaches of incremental learning is carried out. From this study, by applying voting rule to incremental learning a new approach is proposed. Further, the comparison between existing Fuzzy rule method and proposed approach is done.

KEYWORDS

Incremental learning; concept drift; class imbalance; intrusion detection; voting rule

1. INTRODUCTION

An intrusion detection system (IDS) is a component of the computer and information security construction. Its main aim is to evolve between normal activities of the system and behavior that can be classified as suspicious or intrusive. IDS's are needed because of the large number of incidents reported increases every year and the attack techniques are always improving. IDS perspectives can be divided into two main categories: misuse or anomaly detection. The misuse detection approach assumes that an intrusion can be detected by matching the belonging activity with a set of intrusive patterns. Examples of misuse detection contain keystroke monitoring, expert systems, and state transition scrutiny. Misuse detection systems assume that an intrusion should digress the system behavior from its normal pattern. This a anomaly detection approach can be implemented using neural networks, statistical methods, predictive pattern generation and association rules among others procedure.

The rest of this paper is organized as follows. Section 2 gives brief idea about literature survey and problem analysis. Section 3 includes an analysis of previous work with proposed work and result comparison. Section 4 includes conclusion of overall work.

2. LITERATURE SURVEY AND PROBLEM ANALYSIS

Incremental learning has recently attracted growing attention from both academia and industry. Incremental learning is a useful and practical expertise of learning new data over time.

2.1. Incremental learning

Learning incrementally is not only useful because it allows us to refine our models/classifiers over time. There are two issues: First, many of today's data-intensive computing applications necessitate the learning algorithm to be able of incremental learning from large-scale dynamic stream data, and to build up the knowledge base over time to benefit future learning and decision-making process. Second, from the machine intelligence perspective, biological intelligent systems are able to learn information incrementally throughout their lifespan, assemble experience, and flourish spatial-temporal associations. Among the recent efforts on incremental learning from knowledge discovery and data inspection points of view, numerous new algorithms and architectures have been developed and successfully applied to non-identical domains. For instance, an incremental linear discriminate scrutiny (ILDA) was proposed in [1] to grasp the inverse of the within-class scatter matrix issue. Based on ILDA, a new algorithm, namely GSVDILDA, the generalized singular value spoilage LDA, was proposed and successfully applied to the face recognition problem. In [1] and [2], incremental learning for autonomous navigation systems was introduced. A system named Swift File was proposed to help disparate users to tabulate their e-mail messages into folders, which can be dynamically adjusted according to users' mailing habits. Some other works on incremental learning and its applications contains the incremental learning fuzzy neural (ILFN) network for fault detection and classification, incremental learning for multi-sensor data fusion, incremental genetic learning for data organization, incremental semi-supervised learning, incremental learning for human-robot interaction, and others.

Table 1 . Requirements for incremental learning

Requirement	Description
R1	One pass learning
R2	Learn new knowledge
R3	Preserve previous knowledge
R4	Limited processing

Table 1disputes regarding the requirements of incremental learning in the community. For instance, in [3] and [4], whether the previous data can be retrieve by the current learning process in the scenario of incremental learning was deliberate. Besides, in [5], whether the incremental learning should be inspire to handle the unexpected emergent new class was discussed. Recently, it was presented in [6] and [7] that incremental learning should be capable of learning the new information and retaining the previously obtained knowledge, without having access to the previously seen data. An incremental learning algorithm must learn the new information, and keep previously acquired knowledge, without having access to earlier seen data [8], which then elevates the stability plasticity dilemma [9].

Incremental learning of new classes becomes even more challenging if new classes are also unbalanced. Unbalanced (or, also referred to as "imbalanced") data transpire when one or more classes are heavily underrepresented in the training data, a common phenomenon in real world machine learning applications [10]. Unbalanced data are usually grasped by over sampling the minority class or under sampling the majority class data, the most successful implementation of which is the Synthetic Minority Oversampling Technique (SMOTE). Ensemble-based algorithms, in specific variations of AdaBoost, have also been applied in learning class imbalance through cost-sensitive measures for over and under sampling mechanisms for choosing training data. SMOTEBoost; AdaCost; DataBoost and RAMOBoost are examples of such techniques. While all of the aforementioned algorithms are capable to address unbalanced data (with varying levels of success), none of them are well suited for incremental learning. For

incremental learning of new classes there are successful algorithms for incremental learning of new classes, such as Learn++.NC, and for learning from unbalanced data, such as SMOTE, a logical method for incremental learning of new underrepresented classes is a proper integration of such techniques. It is most important to notice that the imbalanced learning over data streams has also attracted significant growing attention in the community.

Intrusion detection is the approach of examine anomalous activity [11]. It includes, supervising and inspecting the events occurring in the network in order to notice malicious activity. Due to the rapid growth in Internet technology, IDS has been gain interesting field of research for the past few decades. The existing IDS are suffering from the provocations like high false alarms and low accuracy, which further not capable to identify an attack correctly. These provocations can be overcome by making IDS intelligent utilized machine learning techniques. Intrusion detection is an area extending in relevance as more and more sensitive data are stored and organized in networked systems [12]. An intrusion detection system (IDS) monitors networked devices and looks for anomalous or malicious action in the patterns of activity in the audit stream. A comprehensive IDS needs a significant amount of human expertise and time for evolution. Data mining based IDSs does not require specific knowledge yet provide better performance. The goal of an ID is to recognize malicious traffic [13]. In order to complete this, the IDS observes all incoming and outgoing traffic. There are various approaches on the implementation of IDS. Among those, two are the most famous: Anomaly detection is based on the detection of traffic anomalies. The deviation of the monitored traffic from the normal activity is measured. Various implementations of this approach have been introduced, based on the features used for measuring traffic profile deviation. Misuse- looks for patterns and signatures of already known attacks in the network traffic. A interminable updated database is usually used to store the signatures of known attacks [14, 15]. The way this technique deals with intrusion detection resembles the way that anti-virus software operates.

2.2. Voting rule

The effective combining rules [16, 17, 19] contains geometric average rule (GA rule), arithmetic average rule (AA rule), median value rule (MV rule), majority voting rule (MajV rule), Borda count rule (BC rule), max and min rule, weighted average rule (Weighted AA rule) and weighted majority voting rule (Weighted MajV rule) , which are encapsulate below.

GA Rule

GA rule finds P(Ym | xt) to reduce the average Kullback–Leibler (KL) separation among probabilities.

$$D_{ab} = \frac{1}{L}\sum_{n=1}^{L} D_n \tag{1}$$

Where,

$$D_n = \sum_{m=1}^{C} P(Y_n|x_t) \ln \frac{P(Y_n|x_t)}{P_j(Y_n|x_t)} \tag{2}$$

Taking Lagrange multipliers and considering $\sum Cm=1\ P(Ym|xt)= 1$, the optimization of (1) with respect to P(Ym | xt) gives us:

$$P(Y_m|x_t) = \frac{1}{A}\prod_{n=1}^{L}(P_n(Y_m|x_t))^{\frac{1}{L}} \tag{3}$$

where, A is a class-independent number.

Based on (3), GA rule divine the testing instant xt to the class identity label that maximizes the product of Pn (Ym | xt).
GA Rule:

$$x_t \rightarrow Y_m \text{ satisfy } \max_{Y_m} \prod_{n=1}^{L} P_n(Y_m|x_t) \tag{4}$$

AA Rule

Instead of using (2), one can also define the probability distance by an alternative KL separation as follows:

$$D_n = \sum_{m=1}^{C} P_n(Y_m|x_t) \ln \frac{P_n(Y_n|x_t)}{P(Y_n|x_t)} \quad (5)$$

Substituting (5) into (1), one can get

$$P(Y_m|x_t) = \frac{1}{L}\sum_{n=1}^{L} P_n(Y_m|x_t) \quad (6)$$

Therefore, the AA rule can be explained as finding the maximum value of the arithmetic average of Pn (Ym | xt).

AA Rule:

$$x_t \rightarrow Y_m \text{ satisfy } \max_{Y_n} \frac{1}{L}\sum_{n=1}^{L} P_n(Y_m|x_t) \quad (7)$$

MV Rule

In the condition of probability outliers of Pn (Ym | xt), the AA rule may lead to poor combination performance since the outliers will control the voting approach. In such a case, the MV rule will predict the final class label with the maximum median value.

MV Rule:

$$x_t \rightarrow Y_m \text{ satisfy } \max_{Y_n}\{\text{median}(P_n(Y_m|x_t))\} \quad (8)$$

MajV Rule

In addition to the soft type rules like GA and AA, MajV rule is a hard type ensemble strategy. Each and every individual classifier directly predicts the class label of the testing sample, and then only MajV rule simply outputs the final predicted label as the one that accepts more votes from the individual classifiers across all classes. When multiple class labels receive the same number of maximum counts, a random class label among them can be selected. It is important to have predicted label from each individual classifier may be obtained from posterior class probabilities. For instance, each net in an ensemble of neural networks first outputs the posterior class probabilities, which it predicts the class label. Then, MajV rule counts the votes from these nets.

MajV Rule:

$$x_t \rightarrow Y_m \text{ satisfy } \max_{Y_n} \sum_{n=1}^{L} \Delta_n(Y_m|x_t) \quad (9)$$

Where,

$$\Delta_n(Y_m|x_t) = \begin{cases} 1, & \text{if } x_n(x_t) = Y_n \\ 0, & \text{otherwise} \end{cases}$$

Max Rule

Max rule is based on the information provided by the maximal value of Pn (Ym | xt) over all potential class labels. Unlike the AA rule which is based on the mean value of Pn (Ym | xt), Max rule is more like a winner-take-all style of voting.
Max Rule:

$$x_t \rightarrow Y_m \text{ sati sfy } \max_{Y_n}\{\max_{n}(P_n(Y_m|x_t))\} \quad (10)$$

Min Rule

Similar to the Max rule, Min rule is based on the idea to vote the final predicted class label have the maximal of the minimal values of Pn (Ym |xt) across all potential class labels.

Similar to (10), Min rule can be defined as follows.

Min Rule:

$$x_t \rightarrow Y_m \text{ satisfy } \max_{Y_n}\{\min_n(P_n(Y_m|x_t))\} \quad (11)$$

BC Rule

The BC rule is based on the categorized order of class labels provided by separate Pn (Ym |xt). Based on the classifier output, each classifier ranks all the potential class labels. For a C class problem, the pth categorized candidate collects (C − p) votes for the final voting systems. Finally, the class label that collects most of the votes will be the final predicted result.
The BC rule can be defined as follows.
BC Rule:

$$x_t \rightarrow Y_m \text{ satisfy } \max_{Y_n}\sum_{n=1}^{L}\Omega_n(Y_m|x_t) \quad (12)$$

where Ωn (Ym |xt) = C −p if classifier h n categorized xt in the pth position for class label Yn, and C is the number of classes.

Weighted rules

In order to reflect different contributions from non-identical classifiers, a weight coefficient can be introduced to each individual classifier in different of the aforementioned methods.
Weighted AA Rule:

$$x_t \rightarrow Y_m \text{ satisfy } \max_{Y_n}\frac{1}{L}\sum_{n=1}^{L}\omega_n.P_n(Y_m|x_t) \quad (13)$$

Weighted MajV Rule

$$x_t \rightarrow Y_m \text{ satisfy } \max_{Y_n}\sum_{n=1}^{L}\omega_n.\Delta_n(Y_m|x_t) \quad (14)$$

where*ωn*is a weight coefficient for classifier h n : ωn≥ 0 and ∑Ln =1 ωn= 1.

SSC Rule

This is a combination method influenced by signal strength concept [14] is used to merge the output of each individual classifier to support the final decision-making procedure in classification.

Procedure:

1. Apply the testing instance xt to each individual classifier h n, and provides the decision profile Pd (Ym | xt), where Ym = 1, . . . ,C is the potential class identity label.

2. Based on each column of the Pd(Ym| xt) in the decision profile, compute the signal strength SYm, the signal strength with direction S1Ym , and uncertainty degree NYmfor each class identity label.

$$S_{Ym} = |P_d(Y_m|x_t) - 0.5| \quad (15)$$

$$S1_{Ym} = P_d(Y_m|x_t) - 0.5 \quad (16)$$

$$N_{Ym} = 0.5 - S_{Ym} \quad (17)$$

Where $S_{Ym} \in [0, 0.5]^L$, $S1_{Ym} \in [-0.5, 0.5]^L$ and $N_{Ym} \in [0, 0.5]^L$

1. Compute S_{Ym} and $S1_{Ym}$

$$\beta_{Ym} = \frac{S_{Ym}}{N_{Ym}} \tag{18}$$

$$\beta 1_{Ym} = \frac{1}{1+e^{-\alpha\beta_{Ym}}} - 0.5 \tag{19}$$

where $\beta_{Ym} \in [0, +\infty]^L$, $\beta 1_{Ym} \in [0, 0.5]^L$and α is a parameter used to regulate the sensitivity level of each voting classifier for its contribution to the final decision.

2. Compute $\beta 1_{out}(Y_m)$ and $S1_{out}(Ym)$

$$\beta 1_{out}(Y_m) = \frac{\sum_{p=1}^{L} \beta 1_{Ym}(p) S1_{Ym}(p)}{\sum_{p=1}^{L} \beta 1_{Ym}(p) N1_{Ym}(p)} \tag{20}$$

$$S1_{out}(Ym) = \frac{\beta 1_{out}(Y_m)}{2(1+|\beta 1_{out}(Y_m)|)} \tag{21}$$

Where $\beta 1_{out}(Y_m) \in \aleph$ and $S1_{out}(Ym) \in [-0.5, 0.5]$

3. Compute the final voting probability $P(Y_n|x_t)$

$$P(Y_m|x_t) = S1_{out}(Ym) + 0.5 \tag{22}$$
Where $P(Y_m|x_t) \in [0,1]$

4. Final decision

The predicted class identity label Ym

$$x_t \rightarrow Y_m \text{ satisfy } \max_{Y_n} P(Y_m|x_t) \tag{23}$$

3. PROPOSED WORK

In this section model of proposed Intrusion Detection System using efficient data mining approach with incremental learning algorithm is introduced. The structure for new proposed technique for intrusion detection system will increase efficiency as compare to existing intrusion detection system.

Intrusion Detection has using simple association algorithm like Apriori and applicable to fuzzy set concept to increase efficiency of association rule mining which is not a good method to originate frequent item set because there is many approaches to build frequent item set in better manner. Another problem we have examined that in fuzzy set theory, it requires multiple analysis before the number of frequent item set exist. It can be very sensitive to the choice of initial analysis. Another drawback is that it does not yield the same result with each run, since the resulting frequent item set depend on the initial unusual assignments. Another disadvantage its not for intrusion detection system because run time efficiency. Data mining efforts in intrusion detection form: Most research centralized on the construction of operational IDSs, rather than on the discovery of new and fundamental insights into the nature of attacks and false positives [18, 20]. It is very common to direct on the data mining step, while the other data mining steps are largely ignored. Much research is based on strong assumptions that complex practical application. Data mining in intrusion detection directs on a small subset of the spectrum of possible applications.

3.1. IDS Architecture

In the proposed method based on the detailed and comprehensive study on data mining based intrusion detection techniques. It is an important thing that a Data Mining does not identify for the intrusions or abnormal activity. What it does is identify patterns within the data that it is processing.

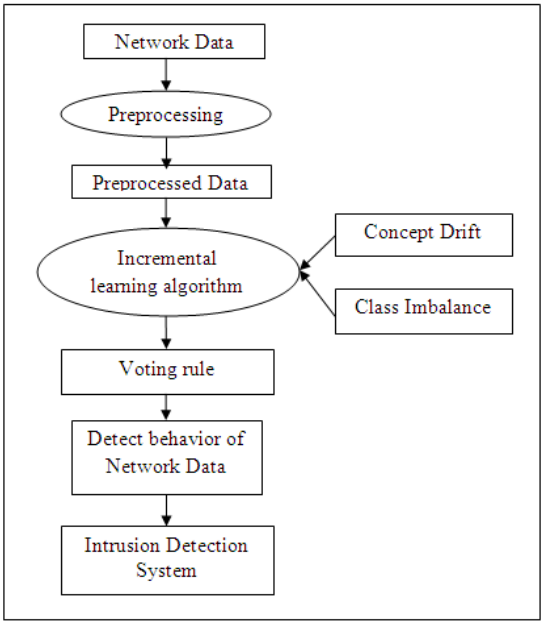

Figure1. Intrusion Detection System (IDS) Architecture

Figure 1 represents that proposed incremental learning approach with concept drift and class imbalance issues. In this technique, it handles two issues very efficiently namely, concept drift and class imbalance. Also, it combines all weights according to signal strength in this way it increases final votes and gives better results.

3.2. Result comparison

This section contains result comparison of existing system and proposed system based on different size of record sets. Comparison of result based on 3 parameters like execution time, memory utilization and CPU utilization.

Approach for Fuzzy rules

This shows the designed approach for automatic generation of fuzzy rules to provide intended learning. The fuzzy rules given to the fuzzy system is done manually, who are given the rules by examining intrusion behavior in network. But, in some cases, it is not easy to generate fuzzy rules manually due to the fact that the input data is very large and also it having number of parameters. But, a few of researches are available in the literature for automatically identifying of fuzzy rules in recent years. Motivated by this fact, here make use of mining methods to identify a better set of rules. There are four rules.

Rule 1-: If Flag contain SYN & FIN then Abnormal Packet
Rule 2-: If Flag is NULL Then it is harmful Packet
Rule 3-: If REJ Packet comes many times then Abnormal.

Rule 4-: If Flag contain combination of SYN, SYN ACK, & ACK Then Normal Packet

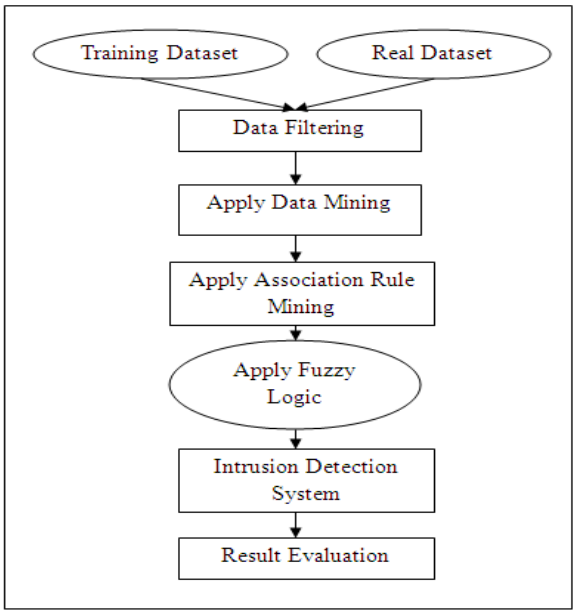

Figure 2. Workflow for Fuzzy Logic

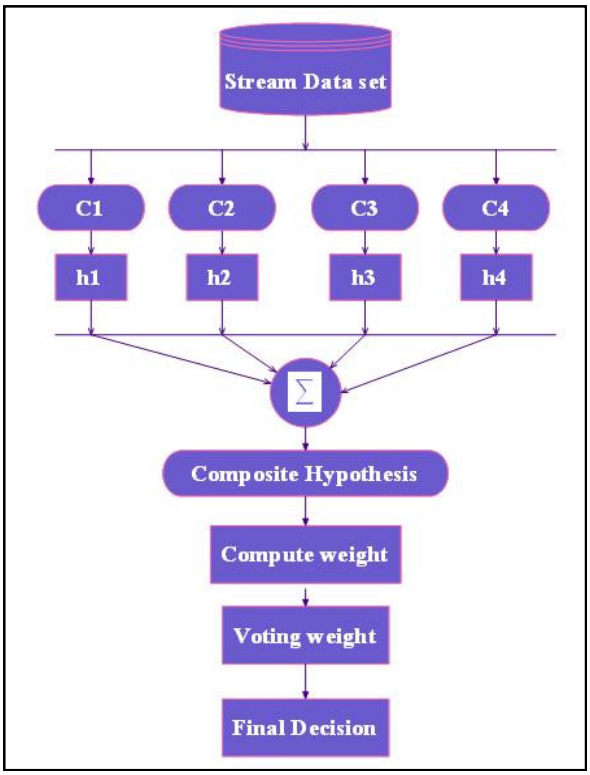

Figure 3. Workflow for Proposed Concept

Figure 2 shows working structure for fuzzy logic which includes some steps like data filtration after that apply some data mining rule then apply fuzzy logic on dataset and finally results are come out.

Figure 3 represents proposed structure for intrusion detection system. Here, on ID data user have to apply incremental learning which solve the issues of concept drift and class imbalance then apply voting rule. In voting rule here SSC voting rule is applied to combine the votes from different classifier which is based on signal strength of dataset.

Table 1. Accuracy for different size record sets

Sr. No.	Instances	Fuzzy logic (in %)	Proposed Algorithm (in %)
1	1000	89.56	99.39
2	5000	90.45	98.84
3	10000	96.88	99.29
4	15000	96.94	99.56
5	20000	91.67	97.34
6	25000	95.78	98.45
7	30000	94.67	99.48
8	50000	96.89	99.56

Table 1 showsaccuracy for different size record sets for fuzzy logic and proposed system and it measures in percentage. It shows that for every record sets proposed system gives better results than existing one. In case of 1000 record set existing system gives 89.56 % results and proposed system gives 99.39% result.

Table 2. Execution time

Sr. No.	Instances	Fuzzy logic	Proposed Algorithm
1	1000	12 Sec	10 Sec
2	5000	27 Sec	26 Sec
3	10000	32 Sec	29 Sec
4	15000	39 Sec	35 Sec
5	20000	48 sec	44 sec
6	25000	59 Sec	53 Sec
7	30000	72 Sec	59 Sec
8	50000	85 Sec	66 Sec

Table 2 shows Execution time for different size record sets for fuzzy logic and proposed system and it measures in seconds. It shows that for every record sets proposed system gives better outcomes than existing one. In case of 1000 recordset existing system requires 12 sec and proposed system requires only 10 sec with better results than existing system.

Execution Time: - The execution time is considered the time that an algorithm takes to produce results. It is used to draw the throughput of an algorithm. It specify the time of algorithm.

CPU Utilization: - The CPU utilization is the time that a CPU is dedicated only to the particular process of calculations. It inspects the load of the CPU. The more CPU time is used in the execution process, the higher is the load of the CPU.

Table 3 shows CPU utilization for different size record sets for fuzzy logic and proposed system and it measures in percentages. It shows that for each and every record sets proposed system gives better outcomes than existing one. For 1000 record set existing system having 60% load on CPU and proposed system having only 58% load on CPU.

Table 3. CPU Utilization

Sr. No.	Algorithm	CPU Utilization (in %)
1	Fuzzy logic	60
2	Proposed Algorithm	58

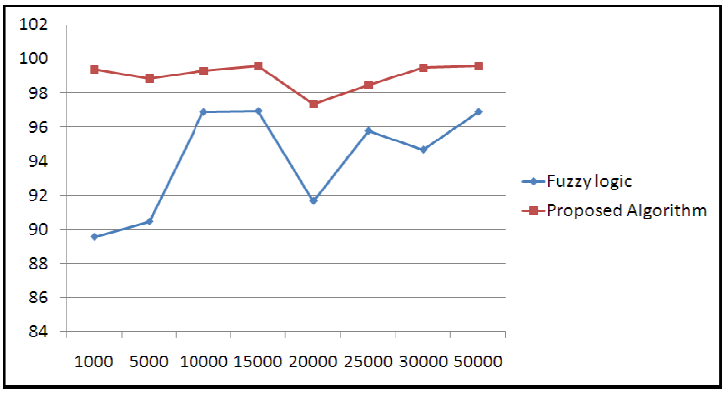

Figure 4. Accuracy for 1-50 thousands record sets

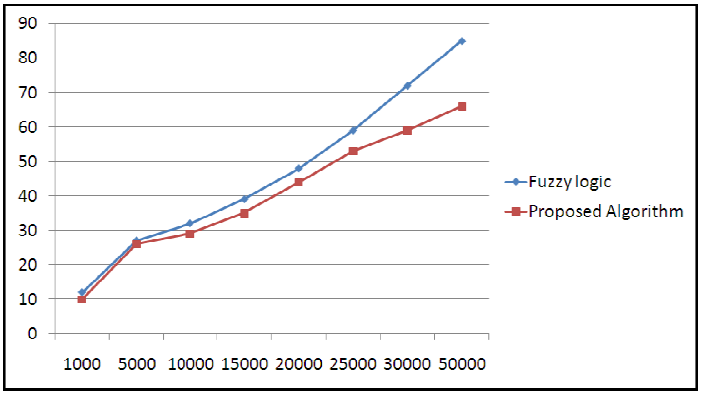

Figure 5. Memory utilization for 1-50 thousands record sets

Figure 4, 5 and 6 shows accuracy, memory utilization and CPU utilization respectively for 1-50 thousands record sets for fuzzy logic and proposed system. . It shows that for each and every record sets proposed system gives better outcomes than existing one. In case of incremental data set there are two issues concept drift and class imbalance which are easily handled with better results by using proposed system for intrusion detection in network data.

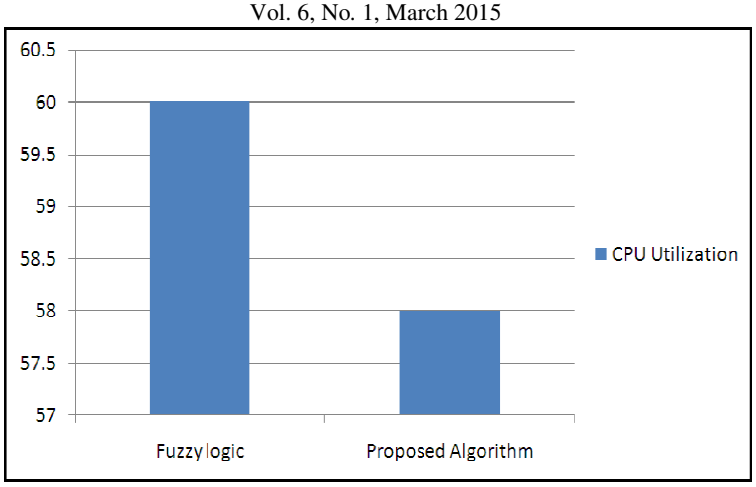

Figure 6. CPU Utilization for 1-50 thousands record sets

4. CONCLUSIONS

In this paper, different method of incremental learning are discussed focusing on the two main issues of concept drift and class imbalance to improve the performance of Intrusion Detection System for network data. The proposed approach is discovered by combining voting rule which is based on signal strength of dataset in incremental learning approach which increases accuracy with reduced time than existing method and less CPU utilization. The effectiveness of proposed approach is justified by comparing the results with existing technique.

REFERENCES

1. Haibo He, Sheng Chen, Kang Li AndXinXu, "Incremental Learning From Strem Data", IEEE Transaction On Neural Networks, Vol.22 No. 12, December 2011.
2. H. Zhao And P. C. Yuen, "Incremental Linear Discriminant Analysis For Face Recognition," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., Vol. 38, No. 1, pp. 210–221, Feb 2008.
3. J. R. Millan, "Rapid, Safe, And Incremental Learning Of Navigation Strategies," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., Vol. 26, No. 3, pp. 408–420, Jun. 1996.
4. G. Y. Chen And W. H. Tsai, "An Incremental-Learning-By-Navigation Approach To Vision-Based Autonomous Land Vehicle Guidance In Indoor Environments Using Vertical Line Information And Multiweighted Generalized Hough Transform Technique," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., Vol. 28, No. 5, pp. 740–748, Oct. 1998.
5. R. B. Segal And J. O. Kephart, "Incremental Learning In Swiftfile," In Proc. 17th Int. Conf. Mach. Learn., 2000, pp. 863–870.
6. G. G. Yen And P. Meesad, "An Effective Neuro-Fuzzy Paradigm For Machinery Condition Health Monitoring," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., Vol. 31, No. 4, pp. 523–536, Aug. 2001.
7. J. Su, J. Wang, And Y. Xi, "Incremental Learning With Balanced Update On Receptive Fields For Multi-Sensor Data Fusion," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., Vol. 34, No. 1, pp. 659–665, Feb. 2004.
8. S. U. Guan And F. Zhu, "An Incremental Approach To Genetic-Algorithms Based Classification," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., Vol. 35, No. 2, pp. 227–239, Apr. 2005.
9. M. Pardowitz, S. Knoop, R. Dillmann, And R. D. Zollner, "Incremental Learning Of Tasks From User Demonstrations, Past Experiences, And Vocal Comments," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., Vol. 37, No. 2, pp. 322–332, Apr. 2007.
10. A.Sharma, "A Note on Batch and Incremental Learnability," J.Comput. Syst. Sci., vol. 56, no.3, pp. 272=276, Jun 1998
11. MithcellRowton, "Introduction to Network Security Intrusion Detection", December 2005.

12. Noel, S., Wijesekera, D., And Youman, C., "Modern Intrusion Detection, Data Mining, And Degrees Of Attack Guilt", In D. Barbarà And S. Jajodia (Eds.), Applications Of Data Mining In Computer Aecurity, Kluwer Academic Publishers, Boston, MA, 2002, pp. 2-25.
13. M. D. Muhlbaier, A. Topalis, And R. Polikar, "Learn++.NC: Combining Ensemble Of Classifiers With Dynamically Weighted Consult-And-Vote For Efficient Incremental Learning Of New Classes," IEEE Trans. Neural Netw., Vol. 20, No. 1, pp. 152–168, Jan. 2009.
14. Haibo He, Yuan cao, "SSC: A Classifier Combination Method based on signal Strength", IEEE Trans. Neural Netw. And Learning systems, vol. 23, no. 7, pp. 1100–1117, July. 2012.
15. Roshani Ade, Dr. P. R. Deshmukh, "Classification of students using psychometric tests with the help of incremental naïve baiyes algorithm", IJCA, Vol.89, No. 14, pp.27-31, March 2014.
16. Roshani Ade, "Incremental Learning in Students Classification System with Efficient Knowledge Transformation", IEEE Conference on PDGC, Dec 2014.
17. Dipali Bhosale, Roshani Ade, "Feature Selection based Classification using Naive Bayes, J48 and Support Vector Machine", International Journal of Computer Applications (0975 – 8887) Volume 99– No.16, August 2014
18. R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments", IEEE Trans. Neural Netw., vol. 22, no. 10, pp. 1517-1531, Oct. 2011.
19. N. Littlestone and M.Warmuth, " Weighted majority algorithm", Information and Computation, vol. 108, pp. 212-261, 1994.
20. Ade, Roshani, and P. R. Deshmukh, "An incremental ensemble of classifiers as a technique for prediction of student's career choice", ICNSC, 2014.