Neural Visualization of Network Traffic Data for Intrusion Detection

EMILIO CORCHADO[1], and ÁLVARO HERRERO[2]

[1]Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced s/n, 37008, Salamanca, Spain
escorchado@usal.es

[2]Department of Civil Engineering, University of Burgos
C/ Francisco de Vitoria s/n, 09006, Burgos, Spain
Tel: +34 947 259513; Fax: +34 947 259395
ahcosio@ubu.es

Abstract **-** This study introduces and describes a novel Intrusion Detection System (IDS) called MOVCIDS (MObile Visualization Connectionist IDS). This system applies neural projection architectures to detect anomalous situations taking place in a computer network. By its advanced visualization facilities, the proposed IDS allows providing an overview of the network traffic as well as identifying anomalous situations tackled by computer networks, responding to the challenges presented by volume, dynamics and diversity of the traffic, including novel (0-day) attacks. MOVCIDS provides a novel point of view in the field of IDSs by enabling the most interesting projections (based on the fourth order statistics; the kurtosis index) of a massive traffic dataset to be extracted. These projections are then depicted through a functional and mobile visualization interface, providing visual information of the internal structure of the traffic data. The interface makes MOVCIDS accessible from any mobile device to give more accessibility to network administrators, enabling continuous visualization, monitoring and supervision of computer networks. Additionally, a novel testing technique has been developed to evaluate MOVCIDS and other IDSs employing numerical datasets. To show the performance and validate the proposed IDS, it has been tested in different real domains containing several attacks and anomalous situations. In addition, the importance of the temporal dimension on intrusion detection, and the ability of this IDS to process it, are emphasized in this work.

Keywords **–** Neural and Exploratory Projection Techniques, Connectionist Unsupervised Models, Computer Network Security, Intrusion Detection, Network Traffic Visualization

1. INTRODUCTION

An attack or intrusion to a network would end up affecting any of the three computer security principles: availability, integrity and confidentiality, exploiting for example the Denial of Service, Modification and Destruction vulnerabilities [1]. One of the most harmful points of attacks and intrusions, increasing the difficulty of protecting computer systems, is the everchanging nature of attack technologies and strategies.

For this reason among others, Intrusion Detection Systems (IDSs) have become a required asset in addition to the computer security infrastructure of most organizations. In the context of computer networks, an IDS can roughly be defined as a tool designed to detect suspicious patterns that may be related to a network or system attack. Intrusion Detection (ID) is then a field focused on the identification of attempted or ongoing attacks in a computer system (Host IDS - HIDS) or network (Network IDS - NIDS). The accurate detection of computer and network system intrusions in real-time has always been an interesting and intriguing problem for system administrators and information security researchers. It could mainly be attributed to the dynamic nature of systems and networks, the creativity of attackers, the wide range of computer hardware and operating systems and so on. Such complexity rises when dealing with distributed network-based systems and insecure networks such as the Internet.

This study introduces an NIDS characterized by the use of an unsupervised connectionist projection technique providing a novel approach based on the visual analysis of the internal structure of the flow of traffic data. Unsupervised learning meets the ID requirements as in a real-life situation there is no target reference with which to compare the response of the network. Additionally, this soft-computing approach is quite useful for identifying unknown or not previously faced attacks, known as 0-day attacks, based on the well-know generalization capability of the Artificial Neural Networks (ANNs).

It is important to note that the authors propose MOVCIDS (MObile Visualization Connectionist Intrusion Detection System) also as a complementary tool to other network security ones, this is, MOVCIDS can work in unison with other defence mechanisms (even if they are IDSs), to provide an intuitive depiction of both normal and anomalous traffic.

The remaining five sections of this study are structured as follows: section 2 contains a brief state of the art of IDSs (mainly visualization-based). Section 3 describes the neural projections techniques applied in this work, while section 4 provides an overview of the proposed IDS, in which each step forming this system is described in detail. Some experimental results are presented and described in section 5; the proposed IDS is tested in some different ways in section 6; authors discuss the considered main advantages of MOVCIDS in section 7 and finally, section 8 puts forward a number of conclusions and pointers for future work.

2. PREVIOUS WORK

ID has been approached from several different points of view up to now; many different intelligent and Soft Computing techniques (such as Genetic Programming [2, 3], Data Mining [4-10], Expert Systems [11, 12], Fuzzy Logic [13, 14], or Neural Networks [15-20] among others) together with statistical [21] and signature verification [22] techniques have been applied mainly to perform a 2-class classification (normal/anomalous or intrusive/non-intrusive). Most of these systems can generate different alarms when an anomalous situation is detected, but they can not provide a general overview of what is happening inside a computer network.

From an opposite point of view, a great variety of visualization-based approaches to ID have been proposed as well [23-34]. In this case, the ID task is enabled by providing a visual depiction of the network or the traffic. Thus, the identification of attacks must be performed through visual features because no alarms are triggered. Visualization tools rely on the human ability to recognize different features and detect anomalies through graphical devices [35]. One of the main advantages is that apart from enabling the anomalies detection, this approach could provide a general snapshot of network traffic. As this study focuses on visualization of network traffic data rather than network structure or topology, previous work only on network data visualization is considered.

Network data are summarized in previous work by:

- **IP addresses**: that is the case of the Galaxy View of NVisionIP [36]. In [37], Border Gateway Protocol data are visualized by a diagram based on IP addresses. A matrix based on IP addresses is proposed as well in [30] to detect the propagation of the Welchia and Sasser. D worms. The Time-based Network Traffic Visualizer [31] combines a matrix display of host IP address and packets timestamp. IP segments are used in NIVA [38] to locate and colour the data.
- **Port numbers**: in [24] the main visualization proposed is based on port and time information. Stacked histograms of aggregate port activity are proposed in [25]. In the case of NVisionIP [36], the previously mentioned Galaxy View is completed by the Small Multiple View, that uses port numbers to visualize the data. By using port numbers and IP addresses, the system proposed in [25] is able to see the penetration and subsequent activity of the Sasser worm.

- **Different measurements of network traffic**: the Multi Router Traffic Grapher [26] shows the incoming/outgoing traffic in Bits per Second while IDGraphs [33] uses the number of unsuccessful connections [39].
- **Alarm data**: generated by different IDSs, such as Snort [40] or StealthWatch IDS [41].
- **Others**: additional kinds of data can be also processed by different visualization tools, such as VIAssist [42] or IDtk [28] that are applied to raw TCP packet data or alerts generated by IDS tools.

In contrast to other security tools, IDSs need to be monitored [43]. So, an IDS can be useless if nobody is looking at its outputs. In keeping with this idea, MOVCIDS goes beyond the state of the art in relation to previously mentioned visualization tools, combining features extracted from packet headers to depict each simple packet by using neural unsupervised methods based on Exploratory Projection Pursuit (EPP) [44] [45] . It provides the network administrator with a snapshot of network traffic, protocol interactions, and traffic volume generally in order to identify anomalous situations. To do so, an unsupervised neural model (see section 3) is applied.

Most of the solutions described in this section use a glyph metaphor [28, 38, 46] to encode information by changing different features (colour, size, opacity, etc.) in addition to the spatial coordinates, while others use traditional representation techniques such as histograms [25, 47, 48], histographs [39] or other graphs [29, 32]. The novel IDS proposed in this work employs the glyph metaphor as well, using different colours and shapes in addition to the spatial coordinates to offer information about the protocol each packet belongs to.

The connectionist visualization approach is not a new one; [34] proposes a visualization based on the information stored in event logs. These events are considered as multidimensional vectors, and a 2D representation of them is obtained by the Self-Organizing Map (SOM) [49], where new (or anomalous) user activities are identified by visual comparison.

From a purely projection of packets standpoint, Principal Component Analysis (PCA) [50], [51], has been also proposed as a visualization tool for analyzing network data [23, 27]. The PCA-based visualization provided in [23] does not enable to distinguish attacks from normal traffic. Furthermore, an explanation of the projection obtained by this technique is not yielded. In [27] PCA is proposed as a complementary tool to interpret the results obtained by a statistical analysis because the visualization does not allow the identification of attacks on its own.

Previous work on this projection approach also includes the application of a visualization tool for intrusion detection [52]. Although some attacks are visually identified in that work by combining visualization and fuzzy feature extraction, explanations about the projection technique and the identification process are not provided.

The novel IDS presented in this study also employs scatterplot matrixes to visualize packet data and provides a proper explanation of the results obtained by projection methods such as PCA (based on the second order statistic, i.e, the variance) and also going further, applying connectionist models based on higher order statistics such as the kurtosis (which is a measure of how pointed a distribution is).

3. UNSUPERVISED CONNECTIONIST PROJECTION ARCHITECTURES

The identification of patterns that exist across dimensional boundaries in high dimensional datasets is a fascinating task [44]. Such patterns may become visible if changes are made to the spatial coordinates. However, an a priori decision as to which parameters will reveal most patterns requires prior knowledge of unknown patterns.

Projection methods project high-dimensional data points onto a lower dimensional space in order to identify "interesting" directions in terms of any specific index or projection. Such indexes or projections are, for example, based on the identification of directions that account for the largest variance of a dataset –as is the case of PCA [50, 51] - or the identification of higher order statistics such as the skew or kurtosis index -as is the case of Exploratory Projection Pursuit (EPP) [44]. Having identified the most interesting projections, the data are then projected onto a lower dimensional subspace plotted in 2D or 3D, which makes it possible to examine its structure with the naked eye. The remaining dimensions are discarded as they mainly relate to a very small percentage of the information or the dataset structure. In that way, the structure identified through a multivariable dataset may be visually analyzed with greater ease. In this work, we take advantage of this dimensionality reduction ability to perform a 2D visualization of the analyzed data (from a 5-dimensional space) through an unsupervised projection model.

Scatterplot matrixes [53] based on projection techniques constitute a useful visualization tool to investigate the intrinsic structure of multidimensional data, enabling experts to see the relations between different components, factors or projections.

3.1. A Variance-based Visualization

PCA is a standard statistical technique for compressing data; it can be shown to give the best linear compression of the data in terms of least mean square error. There are several ANNs or connectionist models which have been shown to perform PCA e.g. [54-56]. This technique describes the variation in a set of multivariate data in terms of a set of uncorrelated variables, in decreasing order of importance, each of which is a linear combination of the original variables. It should be noted that even if we are able to characterize the data with a few variables, it does not follow that an interpretation will ensue.

3.2. Unsupervised Connectionist Visualization for MOVCIDS

Exploratory Projection Pursuit (EPP) [44] is a more recent statistical method aimed at solving the difficult problem of identifying structure in high dimensional data. It does this by projecting the data onto a low dimensional subspace in which we search for data's structure by eye. However, not all projections will reveal this structure equally well. It therefore defines an index that measures how "interesting" a given projection is, and then represents the data in terms of projections maximizing that index.

The first step for EPP is to define which indexes represent interesting directions. "Interestingness" is usually defined with respect to the fact that most projections of high-dimensional data give almost Gaussian distributions [45]. Thus, in order to identify "interesting" features in data, it is appropriate to look for those directions onto which the data-projections are as far from the Gaussian as possible.

Two simple measures of deviation from a Gaussian distribution are based on the higher order moments of the distribution. Skewness is based on the normalized third moment and measures the deviation of the distribution from bilateral symmetry. Kurtosis is based on the normalized fourth moment and measures the heaviness of the tails of a distribution. A bimodal distribution will often have a negative kurtosis and therefore negative kurtosis would signal that a particular distribution shows evidence of clustering.

Because a Gaussian distribution with mean a and variance x is equally interesting than a Gaussian distribution with mean b and variance y - indeed this second order structure can obscure higher order and more interesting structure - then such information is removed from the data ("sphering").

Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [57, 58] is based on Maximum Likelihood Hebbian Learning (MLHL) [57, 59], an EPP connectionist model. CMLHL includes lateral connections [58, 60] derived from the Rectified Gaussian Distribution (RGD) [61]. The RGD is a modification of the standard Gaussian distribution in which the variables are constrained to be non-negative, enabling the use of non-convex energy functions. The CMLHL architecture is depicted in Fig. 1, where lateral connections are highlighted.

Fig.1 << File ***Fig.-01.pdf*** goes here>>

Lateral connections used by CMLHL are based on the mode of the cooperative distribution that is closely spaced along a non-linear continuous manifold. Due to this, the resultant net can find the independent factors of a dataset in a way that captures some type of global ordering.

Considering an N-dimensional input vector (*x*), an M-dimensional output vector (*y*) and with W_{ij} being the weight (linking input j to output i), CMLHL can be expressed as:

Feed-forward step:

$$y_i = \sum_{j=1}^{\mathbf{N}} W_{ij} x_j , \forall i \ . \qquad (1)$$

Lateral activation passing:

$$y_i(t+1) = \left[y_i(t) + \tau(b - Ay) \right]^+ . \qquad (2)$$

Feedback step:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i , \forall j \ . \qquad (3)$$

Weight change:

$$\Delta W_{ij} = \eta . y_i . sign(e_j) | e_j |^p . \qquad (4)$$

Where: η is the learning rate, τ is the "strength" of the lateral connections, b the bias parameter and p is a parameter related to the energy function [57-59].

A is a symmetric matrix used to modify the response to the data whose effect is based on the relation between the distances among the output neurons. It is based on the Cooperative Distribution, but to speed learning up, it can be simplified to:

$$A(i,j) = \delta_{ij} - \cos\left(2\pi\left(i-j\right)/M\right). \qquad (5)$$

Where δ_{ij} is the Kronecker delta.

CMLHL has already proved to successfully perform data visualization. It was initially applied to the artificial vision field [58, 60] and then to some other problems [62-64].

3.3. Self-Organizing Map

The Self-Organizing Map (SOM) [49] was developed as a visualization tool for representing high dimensional data on a low dimensional display. Although it is also based on the use of unsupervised learning, it is not a projection architecture but a topology preserving mapping model using competitive learning instead. A SOM, composed of a discrete array of *L* nodes arranged on an *N*-dimensional lattice, maps these nodes into a *D*-dimensional data space while preserving their ordering. The dimensionality of the lattice (*N*) is normally smaller than that of the data, in order to perform the dimensionality reduction. An example of a trained two-dimensional lattice is shown in Fig. 2.a. Typically, the array of nodes is one or two-dimensional, with all nodes connected to the *N* inputs by an *N*-dimensional weight vector as can be seen in Fig. 2.b.

<< File ***Fig.-02a.pdf*** goes here>> << File ***Fig.-02b.pdf*** goes here>>

Fig. 2.

The SOM can be viewed as a non-linear extension of PCA, where the map manifold is a globally non-linear representation of the training data [65]. The self-organization process is commonly implemented as an iterative on-line algorithm, although a batch version also exists. An input vector is presented to the network and a winning node, whose weight vector is the closest (in terms of Euclidean distance) to the input, is chosen.

So the SOM is a vector quantizer (VQ), and data vectors are quantized to the reference vector in the map that is closest to the input vector. The weights of the winning node and the nodes close to it are then updated to move closer to the input vector. When this algorithm is iterated sufficiently, the map self-organizes to produce a topology-preserving mapping of the lattice of weight vectors to the input space based on the statistics of the training data. This connectionist model is applied here for comparative purposes as it is one of the most widely used unsupervised neural models for visualizing structure in high-dimensional datasets and also applied in the field of IDSs [18].

3.4. Curvilinear Component Analysis

Curvilinear Component Analysis (CCA) [66] is a nonlinear dimensionality reduction method. It was developed as an improvement on the SOM. It tries to circumvent the limitations inherent in some previous linear models such as PCA.

The principle of CCA is a self-organized neural network performing two tasks: a vector quantization of the submanifold in the dataset (input space) and a nonlinear projection of these quantizing vectors toward an output space, providing a revealing view of the way in which the submanifold unfolds. It is shown in Fig. 3.

Fig.3 << File ***Fig.-03.pdf*** goes here>>

As regards its goal, the projection part of CCA is similar to other nonlinear mapping methods; in that it minimizes a cost function based on interpoint distances in both input and output spaces. Quantization and nonlinear mapping are separately performed by two layers of connections: firstly, the input vectors are forced to become prototypes of the distribution using a VQ. Then, the output layer builds a nonlinear mapping of the input vectors by considering Euclidean distances.

4. A MOBILE VISUALIZATION IDS: MOVCIDS

The novel IDS presented in this study is mainly based on the application of an unsupervised connectionist projection model and is designed to process the continuous data flow coming from a computer network. In order to do so, MOVCIDS (MObile Visualization Connectionist Intrusion Detection System) splits massive traffic data into segments and analyze them, thereby providing administrators with an intuitive snapshot to analyze the kinds of events taking place on the computer network. This visualization tool may be defined as an IDS taking full advantage of the previously described model called CMLHL (see section 3.2) and also of the mobile technology, by the use of PDAs, blackberries, cell phones or mobile devices in general.

Fig.4 << File ***Fig.-04.pdf*** goes here>>

To detect anomalous situations, MOVCIDS performs the following steps (Fig. 4), as described in detail in the following sections:

- **1st step.- Network Traffic Capture** (section 4.1): packets travelling along the network are captured.
- **2nd step.- Data Pre-processing** (section 4.2): the captured data are selected and pre-processed. Traffic is selected by taking into account the protocol at transportation level, and a set of features contained in the headers of the captured packets is selected from the raw network traffic.
- **3rd step.- Segmentation** (section 4.3): the data stream is divided into simple segments and accumulated ones (consisting of the addition of several consecutive simple segments). This allows the network administrator to perform a more local and detailed analysis on some suspicious situations, while preserving a general overview of the network traffic.
- **4th step.- Data Analysis** (section 4.4): CMLHL is applied to analyze the data. This connectionist model drives a compact projection, enabling the 2-D visualization of the 5 packet features selected in the 2nd step. Some other unsupervised models (such as PCA and the SOM) have also been applied in this research for comparison purposes.
- **5th step.- Visualization** (section 4.5): the projections of simple and accumulated segments are presented to the network administrator for scrutiny and monitoring. One interesting feature of this IDS is the mobility; the visualization step may be performed on a device different from the one used for the previous four steps. To improve the accessibility of the system, the administrator may visualize the results on any kind of mobile device, enabling informed decisions to be taken anywhere and at any time. Low-size static images are sent from the server to these mobile devices due to their today reduced capabilities.

4.1. Network Traffic Capture Step

The ID process starts when MOVCIDS captures packets travelling along the network by using sniffing techniques. That is, one of the network interfaces is set up in promiscuous mode, gathering all the information travelling along the network. Every single packet is captured and its header information is stored. Some of the fields for the often used TCP/IP protocols and applications are:

- **TTL (Time to Live)**: timer used to track the lifetime of the packet.
- **TOS (Type of Service)**: parameters for the type of service requested.
- **Protocol**: a code identifying the next encapsulated protocol. In the case of IP header, this field identifies the protocol over IP.
- **Source IP address**: IP address from where the source host sent the packet.
- **Destination IP address**: IP address to which the packet is sent.
- **Source port**: port number from where the source host sent the packet.
- **Destination port**: destination host port number to which the packet is sent.
- **Acknowledgment info (control bit and number)**: in the case of TCP, the reception of packets is acknowledged back to the sender.
- **Size**: total packet size in bytes.
- **Timestamp**: the time when the packet was sent.

An alternative to packet sniffing could be Netflow records [67], that must be pre-processed in a different way.

4.2. Data Pre-processing Step

The captured data are selected and pre-processed, as only a reduced set of fields (features) contained in the headers of the captured packets is selected from the raw network traffic.

IDSs have to deal with the practical problem of high volumes of quite diverse data. To overcome this problem, MOVCIDS splits the traffic into different groups, taking into account the protocol (either UDP, TCP, ICMP…) over IP. Among all the implemented network protocols, there are several of them that can be considered quite more dangerous (in terms of the network security), such as the Simple Network Management Protocol (SNMP) [68]. SNMP was identified as one of the top five most vulnerable services by CISCO [69], specially the two first versions of this protocol that are the most widely used at present time. An attack based on this protocol may severely compromise the security of the whole network [1]. SNMP attacks were also listed by the SANS (SysAdmin, Audit, Network, Security) Institute as one of the top 10 most critical internet security threats [70] [71].

Most security tools focus their attention on attacks coming from the internet but attacks are just as likely to come from inside the network as from the outside. However, due to these reasons, the experimental setting of this study is focused on the identification of anomalous situations concerning SNMP. Hence, as SNMP is based on UDP, only UDP traffic has been considered in this work in order to be focused on some special cases.

In the data pre-processing step, the system performs a data selection from the captured information. The following 5 variables of each packet are extracted: source port, destination port, size, timestamp and protocol (each packet is assigned the code of the protocol over TCP/UDP it belongs to) as described in Table 1.

Variable	**Description**	**Type (range)**
Source port	Port number from where the source host sent the packet.	Integer (from 0 to 65535)
Destination port	Destination host port number to which the packet is sent.	Integer (from 0 to 65535)
Size	Total packet size (in bytes).	Integer
Timestamp	The time when the packet was sent. Difference in relation to the first captured packet of the segment (in ms).	Integer
Protocol	All the protocols deployed in the analyzed network have been codified.	Integer

Table 1. Selected packet variables.

As it is shown in the experimental section (section 5), these 5 features allow the identification of the anomalous situations related to the SNMP. Furthermore, this minimal traffic measurement -characterizing network packets by a reduced set of packet header features- allows for high volume networks monitoring saving a lot of computational cost.

4.3. Segmentation Step

Fig. 5 << File ***Fig.-05.pdf*** goes here>>

MOVCIDS splits the pre-processed data stream into:

- Equal *simple segments* (S_x): each simple segment contains all the packets whose timestamp is between the segment initial and final time limit (t_0 and t_1 for S_1). As can be seen in Fig. 5, there is a slight time overlap between each pair of consecutive simple segments. Both the length (time duration) of the simple segments and the time overlap can be set up by the administrator.

- *Accumulated segments* (A_x): each one of these segments contains several consecutive simple ones (Fig. 5). The length of accumulated segments is also configurable.

One of the main reasons for such a partitioning is to present a long-term picture of the evolution of network traffic to the network administrator, allowing the visualization of attacks lasting longer than the length of a simple segment. Additionally, simple segments allow a near real-time processing of traffic. The longer a dataset is, the more distant from real-time the analysis will be, due to the capture delay.

The main reason for overlapping simple segments is that anomalous situations could conceivably take place between simple segment S_x and S_{x+1} (the next segment following S_x). In this case, it would be necessary to consider some packets twice in order to visualize the end of the anomalous situation and the evolution between simple segments. To prevent confusion of the analyst (for example, the same anomalous situation is visualized in two different simple segments), accumulated segments are visualized at the same time. This will lead the network administrator to realize that there is only one anomalous situation being visualized twice.

Fig. 5 shows a sample segmentation by MOVCIDS. In this study the simple segment length is 10 minutes and the overlapping time between consecutive simple segments is 2 minutes. Table 1 describes the segments used in this work, generated through the above mentioned values.

Dataset	Number of packets	Initial time limit (ms)	Final time limit (ms)
S_1	3,122	1	600,000
S_2	3,026	480,000	1,080,000
S_3	3,235	960,000	1,560,000
S_4	9,673	1,440,000	2,040,000
S_5	10,249	1,920,000	2,520,000
S_6	3,584	2,400,000	3,000,000
S_7	3,051	2,880,000	3,480,000
S_8	2,818	3,360,000	3,960,000
…			
A_2	5,553	1	1,080,000
A_3	8,219	1	1,560,000
A_4	17,262	1	2,040,000
A_5	20,410	1	2,520,000
A_6	23,352	1	3,000,000

A_7	25,633	1	3,480,000
A_8	27,970	1	3,960,000
…			
A_{13}	49,647	1	6,360,000

Table 1. Simple and accumulated segments description.

4.4. Data Analysis Step

Once simple and accumulated segments have been built, a connectionist model is applied to analyze them. The data analysis task is based on the use of CMLHL [57, 58] (see section 3.2) to drive a compact 2D or 3D visualization of the 5-dimensional packet data. As it is previously mentioned, CMLHL is able to provide a projection showing the internal structure of a dataset by considering the fourth order statistics (the kurtosis index).

Projection models project data points onto lower dimensions identified as "interesting" directions in terms of any one specific index. Such indexes are, for example, based on the variance of a data set (such as PCA [50], [51]) or higher-order statistics such as skewedness or kurtosis, as in the case of Exploratory Projection Pursuit (EPP [44]).

Kurtosis is based on the normalised fourth moment of the distribution and measures the heaviness of the tails of a distribution. A bimodal distribution will often have a negative kurtosis and therefore negative kurtosis can signal that a particular distribution shows evidence of clustering.

The main advantage of models like MLHL [57, 59] and the related ones such as CMLHL [57, 58] is that by maximizing the likelihood of the residual with respect to the actual distribution, the learning rule (Equation 4) is matched to the Probability Density Function (PDF) of the residual by applying different values of the "p" parameter specified in the learning rule [57, 59].

With the Maximum rule (Equation 4) [57, 59], the weights learn to remove the projections of the data which are furthest from that determined by the parameter p. Thus, to search for clusters in a data set (typified by a PDF with $p > 2$), it can be used Maximum Likelihood learning with $p < 2$ [57, 59], which would result in weights which are removing any projections which make these residuals unlikely. Therefore the clusters would be found by projecting onto these weights.

Then, it is expected that for leptokurtotic residuals (more kurtotic than a Gaussian distribution), values of $p < 2$ would be appropriate, while for platykurtotic residuals (less kurtotic than a Gaussian), values of $p > 2$ would be appropriate.

4.5. Visualization Step

The projection of each segment is presented to the network administrator. MOVCIDS is accessible from any mobile device to give more accessibility to network administrators, enabling permanent mobile visualization, monitoring and supervision of networks.

Fig. 6 << File ***Fig.-06.pdf*** goes here>>

Fig. 6 shows an example of the visualization provided by MOVCIDS on a mobile device. An emulator was used to test the visualization on a mobile platform. Further details concerning the MOVCIDS visualization are described in section 5.

5. RESULTS AND DISCUSSION

This section shows the empirical results obtained by MOVCIDS in facing both normal and anomalous traffic. As previously motivated, the experimental setting of this study is focused on the identification of anomalous situations concerning SNMP. As there is not any publicly available packet-level dataset containing such attacks, we decided to create our own dataset named GICAP-IDS dataset. The main experimental study of MOVCIDS makes use of this dataset.

Among the SNMP anomalous situations we focused on, only port/network scans are contained in publicly available datasets, such as the DARPA dataset [72]. To check the ability of the proposed IDS in facing such attacks through a well known dataset, this section also includes some results in facing port scans contained in the DARPA dataset. Note that this is not a complete benchmark aimed at comparing the performance of MOVCIDS with some other previous IDSs as it can not be carried out in a fair way as visualisation features can not be compared to classification performance.

5.1. GICAP-IDS dataset

The main dataset used in these experiments (GICAP-IDS dataset) was generated ‘made-to-measure’ in a small-size network. In addition to the SNMP packets ("normal" and "anomalous"), the datasets contain traffic related to other protocols, considered as "normal" traffic. As this network was isolated and protected from external attacks, "normal" traffic was known in advance and empirically tested. All the "normal" traffic in the different analyzed datasets was noticed to be depicted in the same way, i.e. parallel straight lines (see results in this section). Apart from "normal" traffic, three different SNMP anomalous situations were generated by means of hacking tools in the network where traffic was collected.

In this section, some snapshots are shown (Fig. 7). Each one of them depicts all the packets contained in the dataset whose projection is shown. MOVCIDS plots the packets in different colours and shapes taking into account the protocol information, leading to an intuitive visualization. In these snapshots, and in general for projection models, the axes forming the projections are combinations of the features contained in the original datasets, as shown in Fig. 7.a. They X and Y axes of the projections are not associated to a unique original feature.

MOVCIDS is focused on the identification of SNMP-related attacks. Thus, three main anomalous situations are distributed throughout the different segments in this study, namely: scans, SNMP community searches and MIB (Management Information Base) information transfers. These situations (described in the following paragraphs) can be very risky on their own and all together (a network scan followed by a SNMP community search and ending with an MIB information transfer) make an SNMP attack from scratch. That is, an intruder gets some of the SNMP managed information without having any previous knowledge about the network being attacked.

SNMP was oriented to manage nodes in the Internet community [68]; it is used to control routers, bridges, and some other network elements, reading and writing a wide variety of information (such as operating system, version, routing tables, default TTL and so on) about these devices. All this information is stored in the MIB, so it can be defined in broad terms as the database used by SNMP to store information about the elements that it controls.

The analyzed datasets contain examples of all the above described anomalous situations. Apart from that, information concerning "normal" traffic is included as well.

In addition to purely SNMP anomalous situations (SNMP community searches and MIB transfers), MOVCIDS also helps network administrators in detecting network/port scans, that can be easily identified by some other security tools (by looking at source/destination IP addresses for example). The main reason is that all these anomalous situations make an SNMP attack from scratch.

5.1.1. Scans

A port scan may be defined as series of messages sent to different port numbers of a host to gain information on its activity status. These messages could be sent by an external agent to find out more about the network services a host is providing. On the contrary, in a network scan the same port is the target for a number of hosts (usually all the hosts in an IP address range). A port scan provides information on where to probe for weaknesses, for which reason scanning generally precedes any further intrusive activity. A network scan is one of the most common used techniques to identify services that might be accessed without permission [25].

In this experimental study, the datasets contain network scans aimed at port numbers 1,434 (registered port assigned to Microsoft-SQL-Monitor, the target of the W32.SQLExp.Worm) and 65,788 (as an example of dynamic or private port).

5.1.2. SNMP Community Search

The unencrypted "community string" can be seen as the SNMP password for versions 1 and 2. An SNMP community search is characterized by the intruder sending SNMP queries to the same port number of different hosts trying to guess the SNMP community string by means of different strategies (brute force, dictionary, etc.) [71]. Once the community string has been obtained, all the information stored in the MIB is available for the intruder.

5.1.3. MIB Information Transfer

This situation is a transfer of some (or all the) information contained in the SNMP MIB, generally through the *get* (or *get-bulk*) command. This kind of transfer is potentially a dangerous situation. However, the "normal" behaviour of a network may include queries to the MIB. This is a situation in which visualization-based IDSs are especially useful; these situations are visualized in a "special" way by the IDS but it is the network administrator responsibility to decide whether it is a "normal" MIB transfer (known by him) or it is not.

5.1.4. Results

A traffic data capture was performed in a network and several segments were generated, as described in Table 1. Some experiments on each different dataset have been carried out. For the sake of brevity, this section shows a comparison of only some of the results obtained through these experiments. The traffic contained in these segments can be roughly described as:

- **S_1**: It only contains "normal" traffic. That is, no anomalous situations are included in this segment to provide a visual sample of how "normal" traffic behaves.

- **S_2**: Apart from "normal" traffic, it contains two network scans (anomalous situations) aimed at port numbers 1,434 and 65,788 of all the machines in an IP address range.
- **S_3**: It contains "normal" traffic and SNMP community searches aimed at port numbers 161, 1,161 and 2,161 of all the machines in an IP address range. Three different community names were used for each one of these port numbers.
- **S_4**: It contains "normal" traffic and an MIB information transfer generated by the *get-bulk* SNMP command.
- **A_2**: As it is a compilation of the traffic contained in segments S_1 and S_2, this segment contains two network scans aimed at port numbers 1,434 and 65,788.
- **A_3**: In addition to the network scans contained in A_2 (aimed at port numbers 1,434 and 65,788), it also contains the SNMP community searches included in S_3.
- **A_{13}**: This is the longest analyzed segment, containing examples of all the anomalous situations previously described: network scans (port numbers 1,434 and 65,788), SNMP community searches (port numbers 161, 1,161 and 2,161) with three different community names and two MIB information transfers.

Fig. 7 shows some examples of how CMLHL performs when applied to simple segments of 10 minutes (see Table 1). The CMLHL parameter values were optimized after a fine-tuning process following the criteria described in previous studies [57, 58, 60]. The values of the parameters used in each case are provided in the figure legends.

Fig. 7.a shows the projection of a simple segment (S_1) containing no anomalous situations. This is then, the way in which CMLHL depicts "normal" traffic, by means of packets evolving in parallel straight directions. Any sign of non-parallel evolution or high concentration of packets is viewed as an anomaly. It can be seen how in Fig. 7.a all the packets (related to "normal" traffic) evolve in "normal" parallel directions over time. After analyzing each packet that is depicted, it was noticed that a certain ordering related to the input variables is preserved in this and other projections. The original dimensions of the dataset are preserved in some sense as indicated in Fig. 7.a. Additionally, Fig. 7.a allows us to identify a temporal disruption in a protocol traffic. As can be seen in this figure, the traffic related to a certain protocol (Group 1 in Fig. 7.a) is interrupted at a certain point. Thus, the network administrator should realize that this protocol stopped working for a while. This requires an in-depth investigation to ascertain the reasons for such an interruption, as it might not be related to an intrusion.

Fig. 7.b (projection of S_4 dataset) shows how the system identifies an anomalous situation related to an MIB information transfer. This situation (Groups 1 and 2 in Fig. 7.b) is identified as anomalous due to its high concentration of packets (in comparison to the "normal" traffic) and its evolution does not fit straight lines as "normal" traffic does.

<< File ***Fig.-07a.pdf*** goes here>> << File ***Fig.-07b.pdf*** goes here>>

<< File ***Fig.-07c.pdf*** goes here>> << File ***Fig.-07d.pdf*** goes here>>

Fig. 7.

The following two snapshots within Fig. 7 show the projections of an 18 minute-long accumulated segment (A_2 - Fig. 7.c) and then a 106 minute-long one (A_{13} - Fig. 7.d). As can be seen, the same network scans can be identified in both datasets in which they are contained (Groups 1 and 2 in Fig. 7.c, and Group 1 in Fig. 7.d). In addition to this network scan, A_{13} also includes three SNMP community searches (Groups 2, 3 and 4 in Fig. 7.d) and two MIB information transfers (Groups 5, 6, 7 and 8 in Fig. 7.d).

As shown in the experiments (Fig. 7 a-d), MOVCIDS enables the network administrator to identify anomalous situations when packet traffic evolves in non-parallel directions to the "normal" one and when the density of packets is much higher than that of "normal" situations. Also empirically, we have noticed that an evolution in parallel lines means "normal" traffic data.

5.1.5. The Importance of Time

Showing the importance of the use of time information in neural projections is the aim of this section. In keeping with this idea, two different datasets are analyzed:

- Dataset 1 contains an MIB information transfer as well as normal traffic.
- Dataset 2 contains three network scans aimed at port numbers 161, 162 and 3750 as well as normal traffic.

To check the importance of time information, variations of each one of the datasets have been used. The difference between the variations is the inclusion or exclusion of time information. Two different figures are shown for each dataset: the ones on the left side (8.a and 8.c) including time information and the ones on the right side (8.b and 8.d) excluding time information as a variable.

<< File ***Fig.-08a.pdf*** goes here>> << File ***Fig.-08b.pdf*** goes here>>

<< File ***Fig.-08c.pdf*** goes here>> << File ***Fig.-08d.pdf*** goes here>>

Fig. 8.

After visually analyzing these results, it can be said that the inclusion of time information (Fig. 8.a) allows the identification of the MIB transfer (Group 1) contained in dataset 1 in a clear way. As can be seen in the right-side figure (Fig. 8.b), the exclusion of time information concentrate the MIB transfer in a line (Group 1), what may hinder the network administrator in the identification of this anomalous situation.

Fig. 8.c allows the identification of the three scans (Groups 3, 4 and 5) contained in dataset 2. Only one (Group 3) of the scans can be identified in Fig. 8.d. As in the previous case, time information allows the clear identification of the MIB transfer (Groups 1 and 2).

It can be concluded that CMLHL is able to deal with time information. Using this information enables the identification of some anomalous situations that would be unidentifiable otherwise.

5.2. DARPA Dataset

This section describes the empirical verification of MOVCIDS involving a port scan attack in one of the standard corpora for evaluation of network intrusion detection: the MIT Lincoln Laboratory DARPA dataset [72].

The well-known DARPA intrusion detection dataset provided the testbed used in this experimental study as it is still the reference network traffic dataset. The DARPA corpus was assembled in 1998 and 1999 to provide a standard to evaluate IDSs, including a variety of known and new attacks buried in a large amount of normal traffic. The corpus was collected from a simulation network that was used to automatically generate realistic traffic, including attempted attacks. The DARPA corpus provides a widely-used benchmark for ID evaluation on network traffic at the packet level. Although some works have raised questions about the accuracy and reliability of this dataset [73, 74], the DARPA dataset still is the standard corpus for evaluation of NIDSs.

In the present study, only TCP traffic was selected from this dataset as most of the attacks (166 out of 174) contained in the 1998 DARPA corpus are based on this protocol. TCP packets contained in a subset of this dataset are characterized by using the set of features that already proved to be effective for the GICAP-IDS dataset, namely: timestamp, source and destination ports, packet size and protocol. As such, TCP network traffic is mapped in a five-dimensional feature space. By summarizing packet information in this reduced set of features, the proposed framework is able to monitor high volume networks. On the other hand, as a result of using only packet header features, this framework is not able to identify attacks concerning the packet payload.

5.2.1. Port Scans

Among all the SNMP anomalous situations we focused on, only port/network scans are contained in the DARPA dataset. To check the ability of MOVCIDS in facing such attacks through a well known dataset, this section comprises an experimental validation on facing port scans contained in the DARPA dataset.

Tests in this study involved a subset of the 1998 DARPA dataset. This subset contains 10 minutes of the traffic (3,730 packets) captured on the Monday of the second week, including the portsweep attack generated on that day. In the DARPA documentation page [72], a portsweep attack is defined as a surveillance sweep through many ports to determine which services are supported on a single host. In this sample of portsweep attack, packets are sent from the host 192.168.1.10 to the 100 first port numbers of the host 172.16.114.50.

5.2.2. Results

In the following figures, due to the high number of protocols in the DARPA dataset, all the packets except those related to the portsweep (in blue and red) are depicted as black dots.

Fig. 9 << File ***Fig.-09.pdf*** goes here>>

Fig. 9 shows that MOVCIDS manages to identify the anomalous situation contained in the analyzed dataset. The portsweep attack (Group 1) is identified due to its non-parallel evolution to the normal traffic. This shows how MOVCIDS is able to detect anomalous situations in a large dataset by splitting high-volume data streams into segments.

6. TESTING MOVCIDS

Testing an IDS is a common way to establish its effectiveness [75]. Up to the present, there have been no specific testing techniques for visualization based IDSs. Thus, to measure the performance of MOVCIDS, we propose a two-fold analysis:

- A novel mutation-based technique (see section 6.1) has been designed and employed. This testing technique is inspired by previous ones [76, 77] but specialized on IDSs relying on numerical packet features. It is based on measuring the evaluated IDS results when confronting unknown anomalous situations, as the identification of 0-day (previously unseen) attacks is a key issue in ID. Some ID strategies (especially those based on attack signatures or patterns) can not easily deal with such attacks.
- A fair comparison between visualization based ID and other ID techniques is not easy to be carried out. Thus, we have focused on the comparison between different projection techniques. As it is said in section 2, there have been different approaches to ID from a visualization point of view, but only few of them [23, 27, 52] perform a packet projection. Section 6.2 provides a comparison between some projection models facing packet visualization for intrusion detection.

6.1. Mutation Testing Technique

In general, a mutation can be defined as a random change. In keeping with this idea, the developed testing technique changes different features of the packets belonging to a known attack trying to generate previously unseen attacks. As it is explained in section 4, this numerical information to be mutated is extracted from the packet headers. The goal is to test the system in real-life situations that differ from those contained in the training dataset and which might be generated by a hacker.

The modifications created by this testing technique lead to real situations by involving changes in aspects such as: attack length (amount of time that an attack lasts), packet density (number of packets per time unit), attack density (number of attacks per time unit) and time intervals between attacks. The mutations can also concern both source and destination ports, varying between the three different ranges of TCP/UDP port numbers: well known (from 0 to 1,023), registered (from 1,024 to 49,151) and dynamic and/or private (from 49,152 to 65,535).

Some restrictions were imposed to these mutations in order to generate as realistic as possible new attacks: time values must range from 1 to the length of the original segment while port numbers must range from 0 to 65,535.

Each one of these modifications generates a new dataset. MOVCIDS was tested by analyzing several mutated datasets obtained by mutating the segments described in section 4.3. For the sake of simplicity, only the projections of two mutated datasets (Fig. 10) are shown in this section:

- Fig. 10.a shows the projection of a mutated version of the A_2 accumulated segment (A_2'). In this case, the mutation implies changes on the destination port numbers of the network scans contained in the original dataset. The network scan, originally (in A_2) aimed at port number 1,434, is now (in A_2') aimed at port number 23,745. In the second network scan, the original destination port number 65,788 is replaced by 45,232 (in A_2'). Additionally, the packet density of these scans has been lowered; each one of the original scans consists of 60 packets which were reduced to 40 packets in A_2'.
- Fig. 10.b shows the projection of another mutated version of the A_2 accumulated segment (A_2''). In this case, only one network scan is kept (scan aimed at port number 1,434 in A_2 is removed) and the packet density of this scan decreased by enlarging the duration of the attack. In the original dataset (A_2), the network scan lasted 61,858 ms, while in this mutated dataset (A_2''), it lasts 247,432 ms. This mutation matches the strategy of an attacker trying to slip by unnoticed.

To check the ability of MOVCIDS in identifying 0-day attacks, these mutated segments (A_2' and A_2'') were projected through the weights previously calculated for the A_2 segment. These projections will then reveal whether MOVCIDS is able to identify previously unseen attacks. As previously stated, the CMLHL-training parameter values for the A_2 segment were: Number of iterations = 100,000, learning rate = 0.03, p parameter = 0.3, and τ parameter = 0.12.

<< File ***Fig.-10a.pdf*** goes here>> << File ***Fig.-10b.pdf*** goes here>>

Fig. 10.

The network scans contained in both A_2' (Groups 1 and 2 in Fig. 10.a) and A_2'' (Group 1 in Fig. 10.b) segments are labelled anomalous due to its non-parallel evolution to normal traffic. In the case of the A_2'' segment, it is less easy to identify the network scan due to its lower density of packets. It was expected, as higher density of packets is a sign of anomaly.

The projections of the mutated datasets (Fig. 10) can be compared with the projection of the original A_2 segment (Fig 7.c). By doing this, we can say that, due to the generalization capability of the neural model it is able to identify previously unseen network scans. Thus, we can conclude that the proposed mutation testing technique positively evaluates MOVCIDS.

6.2. Comparison with Other Unsupervised Connectionist Models

After applying mutant testing, it was decided to compare the CMLHL outcome with that obtained by other well-known statistical and connectionist models such as PCA, CCA and SOM. Several experiments were conducted to apply these models to the analyzed case studies but, for the sake of simplicity, only projections concerning one dataset (A_2) are provided in this section. The computation of mutual distances when applying CCA, that is highly resource demanding, prevents from using a bigger dataset. For this dataset, only the best results (from a visualization point of view) obtained after tuning the models are shown in Fig. 11.

For the SOM, the following options and parameters were tuned: grid size, batch/online training, initialization, number of iterations and distance criterion among others. The used parameter values were: linear initialization, batch training, hexagonal lattice, Gaussian neighbourhood function. The grid size was determined by means of a heuristic formula.

In the case of CCA, some other parameters, such as alpha, lambda, number of epochs and distance criterion were tuned. The final selected parameter values were: standardized Euclidian distance, lambda = 168,850 (default value), alpha = 0.3 and 7 epochs.

<< File ***Fig.-11a.pdf*** goes here>> << File ***Fig.-11b.pdf*** goes here>>

<< File ***Fig.-11c.pdf*** goes here>> << File ***Fig.-11d.pdf*** goes here>>

Fig. 11.

To compare with CMLHL projection, see Fig. 7.c at section 5.2.

The statistical technique known as PCA (see section 3.1) was applied to the A_2 segment (Fig. 11.a). This technique, already used in the field of IDSs [23], failed to detect the anomalous situations (network scans), although the two principal components amount to 99.99% of the data's variance. None of the network scans (Groups 1 and 2 in Fig. 11.a) contained in A_2 are identified as anomalous traffic because in this projection all the packets evolve in parallel lines (which is associated to normal traffic in this work).

Fig. 11.b shows the projection of A_2 obtained by CCA (see section 3.4) using standardized Euclidean distance. The anomalies could be differentiated from normal traffic on the basis of parallel evolution as in the case of CMLHL. Network scans (Groups 1 and 2 in Fig. 11.b) are depicted in a non parallel way to normal traffic. The main difference between the CMLHL projection and the one obtained by CCA is that CMLHL provides a more clear non-parallelism. In the case of CCA, some of the groups containing normal traffic are depicted in a way similar to those containing the network scans. It is then less clear to identify the anomalous situations than in the CMLHL projection.

Finally, the SOM mapping (see section 3.3) of the A_2 segment is depicted in Fig. 11.c and 11.d. As the SOM can not properly deal with a linear growing variable as the timestamp, this information has been removed from the dataset. For visualization purposes, all the packets in the analyzed segment were labelled according to the following classes: C1 for normal traffic, C2 for the network scan aimed at port number 1,434, and C3 for the network scan aimed at port number 65,788. Fig. 11.c depicts the labelled generated map in which the classes identified by each neuron are shown. The number of instances (packets) belonging to each one of the classes identified by the neurons are shown in parentheses. Fig. 11.d shows the associated U-matrix.

It can be clearly seen that the SOM is able to cluster the analyzed segment. The group of neurons near the upper-left corner of the lattice (Group 1 in Fig. 11.d) gather all the packets labelled as C3, that is, all the packets belonging to the network scan aimed at port number 65,788. No other class of traffic is identified by the neighbouring neurons. Other important group can be identified in the middle of the lattice (Group 2 in Fig. 11.d). This group gathers all the packets associated to the other anomalous situation (C2). Additionally, some packets associated to normal traffic are also identified by these neurons. That is, by using the SOM mapping, normal traffic could be identified as belonging to an anomalous situation (false positive). The rest of the normal traffic is identified by some other different neurons. The quality measures associated to this SOM mapping are: quantization error = 0.016 and topographic error = 0.073.

We may also say that some projection models such as PCA, EPP, MLHL, CMLHL or CCA have one important advantage over other unsupervised neural models (such as the SOM) in that they use time as a key variable when analyzing the evolution of the packets in the traffic dataset.

This allows the depiction of packets in an intuitive way, what is pretty valuable in the field of visual inspection of network traffic for intrusion detection.

7. INTERESTING FEATURES OF MOVCIDS

In the field of Computer Network Security, traffic datasets normally have a categorical and/or textual nature and their conversion into a data type to which visualization techniques may be applied is not always obvious. A novel approach is followed by the presented IDS model, where each simple packet is visualized on its own. The model proposed in this paper offers a complete and intuitive visualization of network traffic by depicting each single packet and providing the network administrator with a snapshot of network traffic, protocol interactions, and traffic volume in order to identify anomalous situations.

Knowledge discovery, pattern recognition, data mining and other such techniques, deal with the problem of extracting interesting classifications, clusters, associations and other patterns from data. The existence of laptops, palmtops, handhelds, embedded systems, and wearable computers is making ubiquitous access to a large quantity of distributed data a reality. Advanced analysis of distributed data for extracting useful knowledge is the next natural step in the increasingly interconnected world of ubiquitous and distributed computing. MOVCIDS has been designed to make it accessible from mobile devices enabling permanent mobile visualization, monitoring and supervision of computer networks.

Usually, time information is not used in ANN-based IDSs. On the contrary, it can be employed as one of the inputs to the neural model embedded in MOVCIDS. Time information provides an idea of how the traffic data evolve. It helps to identify anomalous situations by taking into account such aspects as high packet density and temporal evolution in non parallel directions. The evolution of the packets over time can be appreciated in the obtained projections as indicated in Fig 7.a. In this case the time variable is evolving in a very similar direction than the Y-axis, showing the temporal evolution of the packets. It can be also noted that destination port, protocol and size variables are more related with the evolution along the X-axis.

Despite of the fact that the projections obtained by CMLHL can not contain as much information as other graphs do, CMLHL projections are intuitive as they provide a general overview of traffic evolution. By using MOVCIDS, an inexperienced network administrator can identify anomalous situations just having a quick look at the CMLHL projections. One of the key issues of such an advantage is the preservation of the temporal context of packets, as previously mentioned.

Previous work has presented few techniques to test and evaluate NIDSs. In this study, a novel testing technique is proposed to validate visualization IDSs employing numerical datasets.

Some existing IDSs need a "clean" (free of attacks) training dataset. This is not the case of MOVCIDS, which can be trained with a dataset containing known and/or new attacks (as 0-day attacks), due to its generalization capabilities as any connectionist model.

8. Conclusions and Future Work

This research line has presented a novel connectionist projection IDS which offers to network administrators greater accessibility using any mobile device due to its visualization facilities. To deal with the problem of data volume, the network traffic data stream is pre-processed and split into simple and accumulated segments. The presented IDS is capable of identifying anomalous situations by means of temporal visualization of the system response. Using time information allows us to identify some anomalous situations that would be unidentifiable otherwise.

Signature-based IDSs rely on models of known attacks. Thus, the effectiveness of these systems depends on the "goodness" of their models. This is to say, if a model of an attack does not cover all the possible modifications, the performance of the IDS will be greatly impaired. Some IDSs generate different alarms when an anomalous situation occurs, but they can not provide a general overview of what is happening inside a network. This limitation is overcome by visualization-based IDSs.

This research constitutes one of the first attempts to identify anomalous situations through the visualization of packet data. That is, the analysis does not rely on summarized information (such as TCP connections). On the contrary, MOVCIDS analyses the data concerning each single packet. This idea has been probed to be effective by testing it through real traffic datasets.

From the comparison of different statistical and unsupervised neural models, we can conclude that PCA is not able to identify any of the anomalous situations under study. On the contrary, CCA can identify these situations but in a less clear way than CMLHL. Furthermore, CCA is much more resource demanding than CMLHL as CCA needs to compute the pairwise distance matrix for the whole dataset. Although it is not a proper projection model, SOM was included in this comparative study. This model is able to roughly distinguish anomalies from normal traffic but with a low level of accuracy. MOVCIDS helps network administrators to identify one of the most dangerous set of attacks coming from the inside of a network: those related to SNMP.

Finally, we propose the use of MOVCIDS in combination with other security tools (specially other IDSs) to overcome their limitations (e.g: identification of 0-day attacks).

Further work will focus on the application of different learning rules in the Analysis Step and the use of a high performance computing cluster to speed up the Analysis Step, and make it a proper commercial IDS tool.

ACKNOWLEDGEMENTS

This research is partially supported through the Junta de Castilla and León project BU006A08, Business intelligence for production within the framework of the Instituto Tecnológico de Castilla y León (ITCL) and the Agencia de Desarrollo Empresarial (ADE), and the Spanish Ministry of Education and Innovation project CIT-020000-2008-2. The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A., within the framework of the project MAGNO2008 - 1028.- CENIT Project funded by the Spanish Government.

CAPTIONS

Fig. 1. CMLHL: lateral connections between neighbouring output neurons.

a) A plot of the weights of a SOM with a two-dimensional lattice trained on artificial data from a uniform square distribution.

b) Input-output relation.

Fig. 2. SOM architecture.

Fig. 3. CCA mapping.

Fig. 4. MOVCIDS structure. A graphical representation.

Fig. 5. Data stream segmentation. Each dataset is divided into several simple segments.

(e.g. $S_{1,}$ S_2 and so on) and accumulated ones (e.g. $A_{2,}$ A_3, …).

Fig. 6. MOVCIDS sample visualization.

a) CMLHL Projection of S_1

Number of iterations = 100 000, learning rate = 0.03, p parameter = 0.3, and τ parameter = 0.12.

b) CMLHL Projection of S_4

Number of iterations = 100 000, learning rate = 0.03, p parameter = 0.3, and τ parameter = 0.12.

c) CMLHL Projection of A_2

Number of iterations = 100 000, learning rate = 0.03, p parameter = 0.3, and τ parameter = 0.12.

d) CMLHL Projection of A_{13}

Number of iterations = 400 000, learning rate = 0.036, p parameter = 0.4, and τ parameter = 0.1.

Fig. 7. Projections obtained by CMLHL.

a) CMLHL projection of dataset 1 with time information.

b) CMLHL projection of dataset 1 without time information.

c) CMLHL projection of dataset 2 with time information.

d) CMLHL projection of dataset 2 without time information.

Fig. 8. Comparison of CMLHL projections regarding time information.

Fig. 9. MOVCIDS visualization of a DARPA sample dataset.

a) CMLHL Projection of A_2'.

b) CMLHL Projection of A_2''.

Fig. 10. Projections of mutated segments obtained by CMLHL.

a) PCA projection of A_3.

b) CCA projection of A_3.

c) SOM labelled map.

d) associated U-matrix.

Fig. 11. Comparison of A_3 projections.

REFERENCES

[1] J. M. Myerson, Identifying Enterprise Network Vulnerabilities, International Journal of Network Management. 12 (2002) 135-144.

[2] A. Abraham, C. Grosan, and C. Martin-Vide, Evolutionary Design of Intrusion Detection Programs, International Journal of Network Security. 4 (2007) 328-339.

[3] W. Lu and I. Traore, Detecting New Forms of Network Intrusion Using Genetic Programming, Computational Intelligence. 20 (2004) 475-494.

[4] K. Julisch, Data Mining for Intrusion Detection: A Critical Review, Applications of Data Mining in Computer Security, D. Barbará and S. Jajodia, Eds. Kluwer Academic Publishers, 2002, pp. 33-62.

[5] W. Lee and S. J. Stolfo, A framework for constructing features and models for intrusion detection systems, ACM Transactions on Information and System Security (TISSEC), 2000, pp. 227-261.

[6] Y. H. Liao and V. R. Vemuri, Use of K-Nearest Neighbor Classifier for Intrusion Detection, Computers & Security. 21 (2002) 439-448.

[7] W. Lee, S. J. Stolfo, and K. W. Mok, Adaptive Intrusion Detection: A Data Mining Approach, Artificial Intelligence Review. 14 (2000) 533-567.

[8] G. Giacinto, F. Roli, and L. Didaci, Fusion of Multiple Classifiers for Intrusion Detection in Computer Networks, Pattern Recognition Letters. 24 (2003) 1795-1803.

[9] S. Chebrolu, A. Abraham, and J. P. Thomas, Feature Deduction and Ensemble Design of Intrusion Detection Systems, Computers & Security. 24 (2005) 295-307.

[10] L. Cohen, G. Avrahami, M. Last, and A. Kandel, Info-Fuzzy Algorithms for Mining Dynamic Data Streams, Applied Soft Computing. 8 (2008) 1283-1294.

[11] D. E. Denning, An Intrusion-Detection Model, IEEE Transactions on Software Engineering. 13 (1987) 222-232.

[12] T. F. Lunt, IDES: An Intelligent System for Detecting Intruders, Symposium: Computer Security, Threat and Countermeasures, 1990.

[13] A. Tajbakhsh, M. Rahmati, and A. Mirzaei, Intrusion Detection using Fuzzy Association Rules, Applied Soft Computing. 9 (2009) 462-469.

[14] A. Abraham, R. Jain, J. Thomas, and S. Y. Han, D-SCIDS: Distributed Soft Computing Intrusion Detection System, Journal of Network and Computer Applications. 30 (2007) 81-98.

[15] S. Zanero and S. Savaresi, Unsupervised Learning Techniques for an Intrusion Detection System, ACM Symposium on Applied Computing, 2004, pp. 412-419.

[16] E. Corchado, Á. Herrero, and J. M. Sáiz, Detecting Compounded Anomalous SNMP Situations Using Cooperative Unsupervised Pattern Recognition, 15th International Conference on Artificial Neural Networks (ICANN 2005), 2005, pp. 905-910.

[17] Á. Herrero, E. Corchado, and J. M. Sáiz, An Unsupervised Cooperative Pattern Recognition Model to Identify Anomalous Massive SNMP Data Sending, ICNC05, 2005, pp. 778-782.

[18] S. T. Sarasamma, Q. M. A. Zhu, and J. Huff, Hierarchical Kohonenen Net for Anomaly Detection in Network Security, IEEE Transactions on Systems Man and Cybernetics, Part B. 35 (2005) 302-312.

[19] S. Mukkamala and A. H. Sung, Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines, Transportation Security and Infrastructure Protection.(2003) 33-39.

[20] C. Zhang, J. Jiang, and M. Kamel, Intrusion Detection using Hierarchical Neural Networks, Pattern Recognition Letters. 26 (2005) 779-791.

[21] D. J. Marchette, Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint, Springer-Verlag New York, Inc., 2001.

[22] M. Roesch, Snort–Lightweight Intrusion Detection for Networks, 13th Systems Administration Conference (LISA '99), 1999, pp. 229-238.

[23] T. Goldring, Scatter (and Other) Plots for Visualizing User Profiling Data and Network Traffic, 2004 ACM Workshop on Visualization and Data Mining for Computer Security, Washington DC, USA, 2004, pp. 119-123.

[24] C. Muelder, K. L. Ma, and T. Bartoletti, Interactive Visualization for Network and Port Scan Detection, Nineth International Symposium on Recent Advances in Intrusion Detection (RAID 2006), 2006, pp. 265-283.

[25] A. Kulsoom, C. Lee, G. Conti, and J. A. Copeland, Visualizing Network Data for Intrusion Detection, Sixth Annual IEEE Information Assurance Workshop - Systems, Man and Cybernetics (SMC) , 2005, 2005, pp. 100-108.

[26] MRTG: The Multi Router Traffic Grapher. http://www.mrtg.org

[27] K. Labib and V. R. Vemuri, An Application of Principal Component Analysis to the Detection and Visualization of Computer Network Attacks, Annals of Telecommunications. 61 (2006) 218-234.

[28] A. Komlodi, P. Rheingans, A. Utkarsha, J. R. Goodall, and J. Amit, A User-Centered Look at Glyph-Based Security Visualization, IEEE Workshop on Visualization for Computer Security (VizSEC 05), 2005, pp. 21-28.

[29] R. A. Becker, S. G. Eick, and A. R. Wilks, Visualizing Network Data, IEEE Transactions on Visualization and Computer Graphics. 1 (1995) 16-28.

[30] H. Koike, K. Ohno, and K. Koizumi, Visualizing Cyber Attacks Using IP Matrix, IEEE Workshop on Visualization for Computer Security (VizSEC 05), 2005, pp. 91-98.

[31] J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi, Preserving the Big Picture: Visual Network Traffic Analysis with TNV, IEEE Workshop on Visualization for Computer Security (VizSEC 05), 2005, pp. 47-54.

[32] A. Oline and D. Reiners, Exploring Three-Dimensional Visualization for Intrusion Detection, IEEE Workshop on Visualization for Computer Security (VizSEC 05), 2005, pp. 113-120.

[33] P. Ren, Y. Gao, Z. C. Li, Y. Chen, and B. Watson, IDGraphs: Intrusion Detection and Analysis Using Stream Compositing, IEEE Computer Graphics and Applications. 26 (2006) 28-39.

[34] L. Girardin, An Eye on Network Intruder-Administrator Shootouts, 1st Workshop on Intrusion Detection and Network Monitoring, Santa Clara, California, 1999.

[35] C. Ahlberg and B. Shneiderman, Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays, Readings in Information Visualization: using Vision to Think. Morgan Kaufmann Publishers Inc., 1999, pp. 244-250.

[36] K. Lakkaraju, W. Yurcik, and A. J. Lee, NVisionIP: Netflow Visualizations of System State for Security Situational Awareness, 2004 ACM Workshop on Visualization and Data Mining for Computer Security, Washington DC, USA, 2004, pp. 65-72.

[37] S. T. Teoh, K. L. Ma, S. F. Wu, and X. Zhao, Case Study: Interactive Visualization for Internet Security, IEEE Conference on Visualization (Vis 2002), Boston, Massachusetts, 2002.

[38] K. Nyarko, T. Capers, C. Scott, and K. A. Ladeji-Osias, Network Intrusion Visualization with NIVA, an Intrusion Detection Visual Analyzer with Haptic Integration, 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS 2002), 2002, pp. 277-284.

[39] P. Ren, Y. Gao, Z. C. Li, Y. Chen, and B. Watson, IDGraphs: Intrusion Detection and Analysis Using Histographs, IEEE Workshop on Visualization for Computer Security (VizSEC 05), 2005, pp. 39-46.

[40] H. Koike and K. Ohno, SnortView: Visualization System of Snort Logs, 2004 ACM Workshop on Visualization and Data Mining for Computer Security, Washington DC, USA, 2004, pp. 143-147.

[41] K. Abdullah, C. P. Lee, G. Conti, J. A. Copeland, and J. Stasko, IDS RainStorm: Visualizing IDS Alarms, IEEE Workshop on Visualization for Computer Security (VizSEC 05), 2005, pp. 1-10.

[42] A. D. D'Amico, J. R. Goodall, D. R. Tesone, and J. K. Kopylec, Visual Discovery in Computer Network Defense, IEEE Computer Graphics and Applications. 27 (2007) 20-27.

[43] A. Chuvakin, Monitoring IDS, Information Security Journal: A Global Perspective. 12 (2004) 12 - 16.

[44] J. H. Friedman and J. W. Tukey, A Projection Pursuit Algorithm for Exploratory Data-Analysis, IEEE Transactions on Computers. 23 (1974) 881-890.

[45] P. Diaconis and D. Freedman, Asymptotics of Graphical Projection Pursuit, The Annals of Statistics. 12 (1984) 793-815.

[46] R. F. Erbacher, Visual Traffic Monitoring and Evaluation, Conference on Internet Performance and Control of Network Systems II, 2001, pp. 153–160.

[47] F. Mansmann, D. A. Keim, S. C. North, B. Rexroad, and D. Sheleheda, Visual Analysis of Network Traffic for Resource Planning, Interactive Monitoring, and Interpretation of Security Threats, IEEE Transactions on Visualization and Computer Graphics. 13 (2007) 1105-1112.

[48] K. Stockinger, E. W. Bethel, S. Campbell, E. Dart, and K. Wu, Detecting Distributed Scans using High-Performance Query-Driven Visualization, 2006 ACM/ IEEE Conference on Supercomputing, Tampa, Florida, 2006.

[49] T. Kohonen, The Self-Organizing Map, IEEE. 78 (1990) 1464-1480.

[50] H. Hotelling, Analysis of a Complex of Statistical Variables into Principal Components, Journal of Education Psychology. 24 (1933) 417-444.

[51] K. Pearson, On Lines and Planes of Closest Fit to Systems of Points in Space, Philosophical Magazine. 2 (1901) 559-572.

[52] X. Jianqiang, J. E. Dickerson, and J. A. Dickerson, Fuzzy Feature Extraction and Visualization for Intrusion Detection, 12th IEEE International Conference on Fuzzy Systems, 2003, pp. 1249-1254.

[53] N. Elmqvist, P. Dragicevic, and J. D. Fekete, Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation, IEEE Transactions on Visualization and Computer Graphics. 14 (2008) 1539-1148.

[54] E. Oja, A Simplified Neuron Model as a Principal Component Analyzer, Journal of Mathematical Biology. 15 (1982) 267-273.

[55] D. Sanger, Contribution Analysis: a Technique for Assigning Responsibilities to Hidden Units in Connectionist Networks, Connection Science. 1 (1989) 115-138.

[56] C. Fyfe, A Neural Network for PCA and Beyond, Neural Processing Letters. 6 (1997) 33-41.

[57] E. Corchado, D. MacDonald, and C. Fyfe, Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit, Data Mining and Knowledge Discovery. 8 (2004) 203-225.

[58] E. Corchado and C. Fyfe, Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors, International Journal of Pattern Recognition and Artificial Intelligence. 17 (2003) 1447-1466.

[59] C. Fyfe and E. Corchado, Maximum Likelihood Hebbian Rules, 10th European Symposium on Artificial Neural Networks (ESANN 2002), 2002, pp. 143-148.

[60] E. Corchado, Y. Han, and C. Fyfe, Structuring Global Responses of Local Filters Using Lateral Connections, Journal of Experimental & Theoretical Artificial Intelligence. 15 (2003) 473-487.

[61] H. S. Seung, N. D. Socci, and D. Lee, The Rectified Gaussian Distribution, Advances in Neural Information Processing Systems. 10 (1998) 350-356.

[62] E. Corchado, P. Burgos, M. D. Rodriguez, and V. Tricio, A Hierarchical Visualization Tool to Analyse the Thermal Evolution of Construction Materials, CDVE 2004, 2004, pp. 238-245.

[63] E. Corchado, M. A. Pellicer, and M. L. Borrajo, A MLHL Based Method to an Agent-Based Architecture, International Journal of Computer Mathematics (Accepted - In press).(2009)

[64] Á. Herrero, E. Corchado, L. Sáiz, and A. Abraham, DIPKIP: A Connectionist Knowledge Management System to Identify Knowledge Deficits in Practical Cases, Computational Intelligence (Accepted - in press).(2009)

[65] H. Ritter, T. Martinetz, and K. Schulten, Neural Computation and Self-Organizing Maps; An Introduction, Addison-Wesley Longman Publishing Co., Inc., 1992.

[66] P. Demartines and J. Herault, Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets, IEEE Transactions on Neural Networks. 8 (1997) 148-154.

[67] Cisco IOS NetFlow. http://www.cisco.com/web/go/netflow

[68] J. Case, M. S. Fedor, M. L. Schoffstall, and C. Davin, Simple Network Management Protocol (SNMP), IETF RFC 1157, 1990.

[69] Vulnerability Statistics Report, Cisco Secure Consulting, 2000.

[70] S. Institute, The Top 10 Most Critical Internet Security Threats - (2000-2001 Archive), 2001.

[71] S. Northcutt, M. Cooper, K. Fredericks, M. Fearnow, and J. Riley, Intrusion Signatures and Analysis, New Riders Publishing Thousand Oaks,, 2001.

[72] DARPA Intrusion Detection Evaluation. http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.html

[73] M. V. Mahoney and P. K. Chan, An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection, Sixth International Symposium on Recent Advances in Intrusion Detection (RAID 2003), 2003, pp. 220-237.

[74] J. McHugh, Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 Darpa Off-line Intrusion Detection System Evaluation as Performed by Lincoln Laboratory, ACM Transactions on Information and System Security. 3 (2000) 262-294.

[75] M. J. Ranum, Experiences Benchmarking Intrusion Detection Systems, NFR Security Technical Publications, 2001.

[76] G. Vigna, W. Robertson, and D. Balzarotti, Testing Network-Based Intrusion Detection Signatures Using Mutant Exploits, 11th ACM Conference on Computer and Communications Security, Washington DC, USA, 2004, pp. 21–30.

[77] R. Marti, THOR: A Tool to Test Intrusion Detection Systems by Variations of Attacks. Diploma Thesis. ETH Zurich, 2002.

Figure 1.

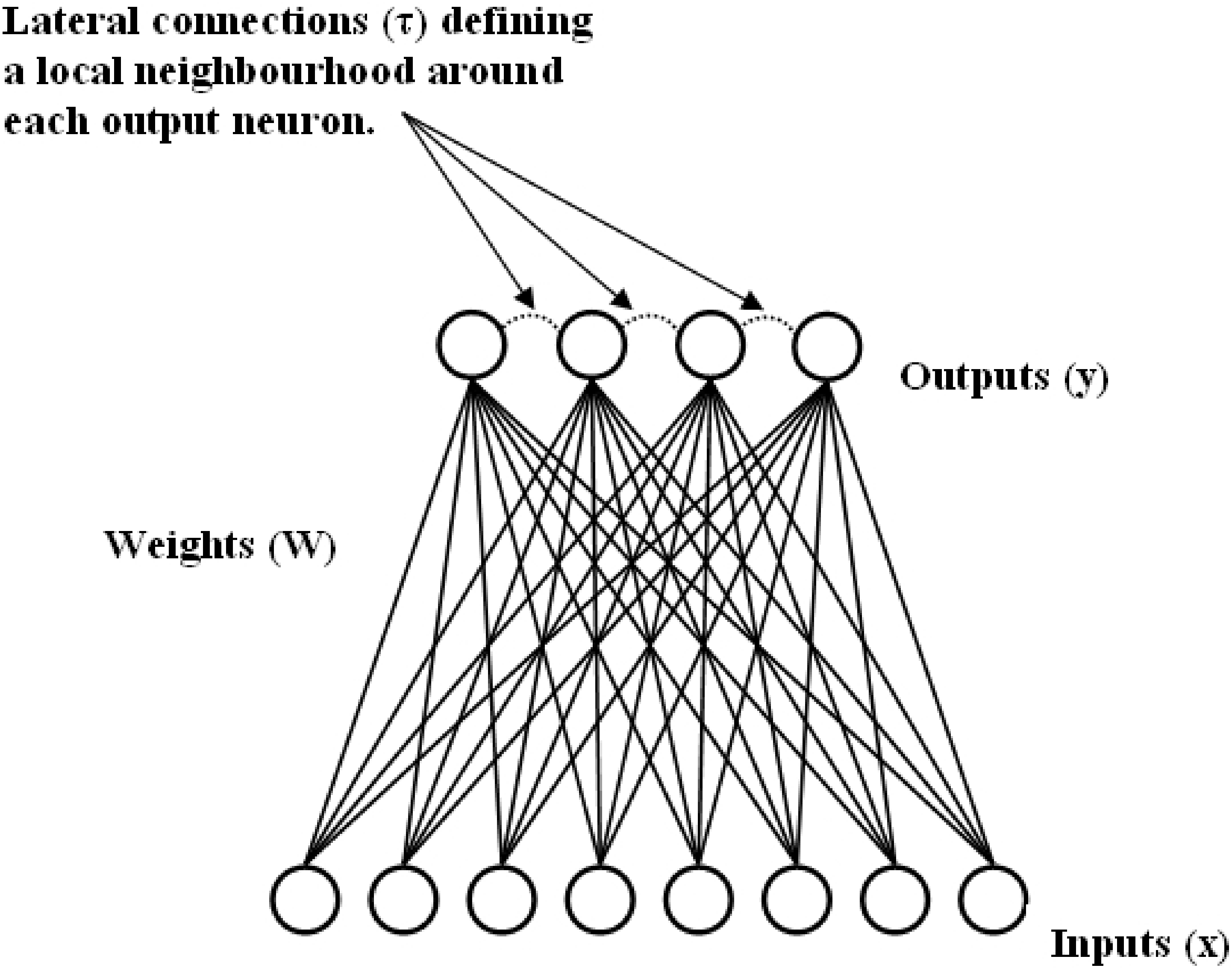

Lateral connections (τ) defining a local neighbourhood around each output neuron.
Outputs (y)
Weights (W)
Inputs (x)

Figure 2a

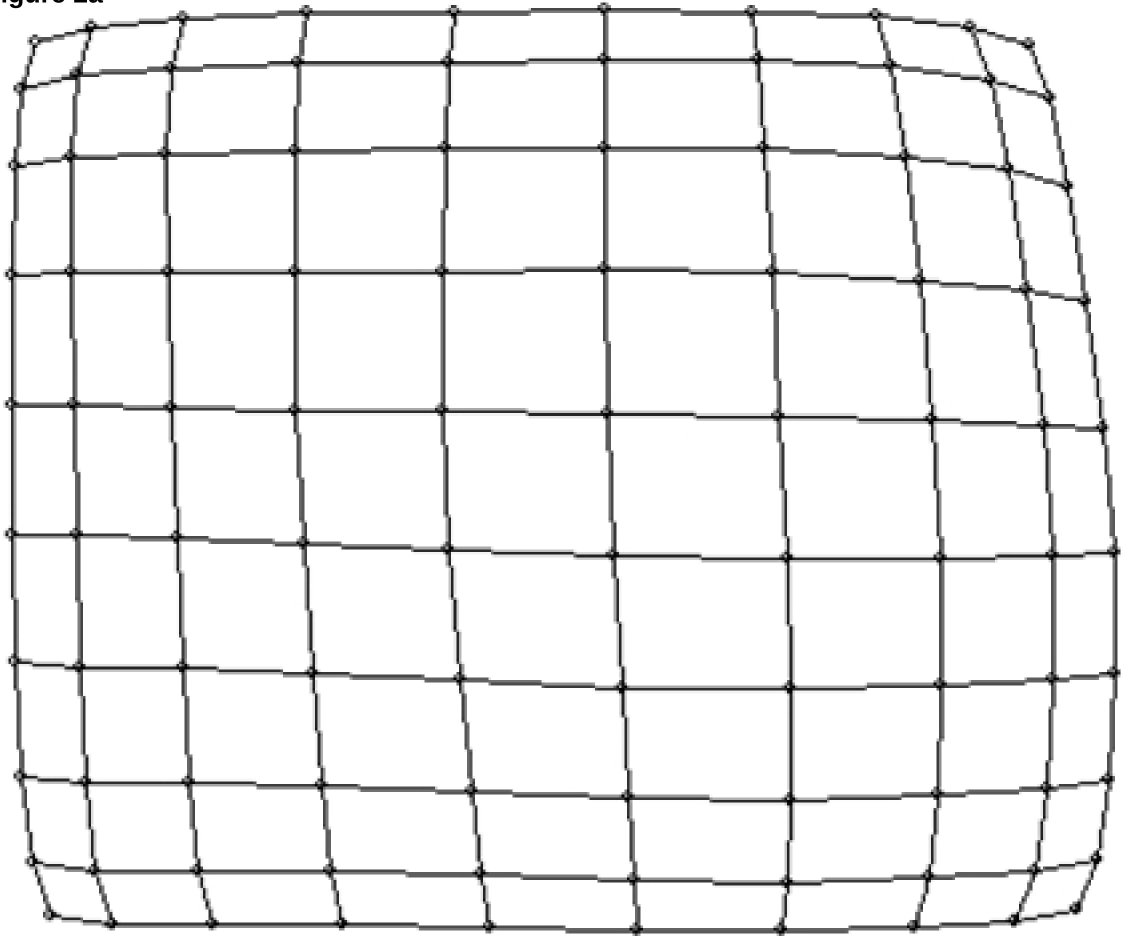


Figure 2b

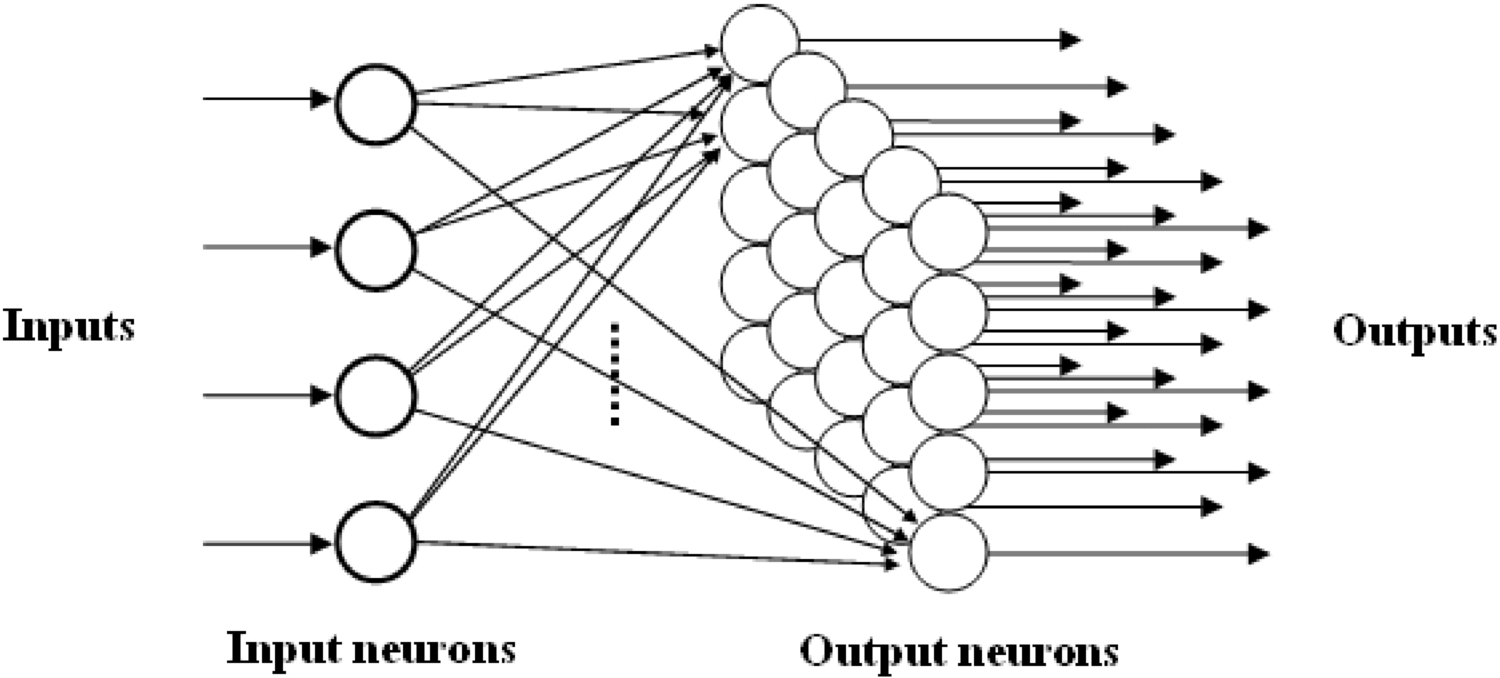


Figure 3

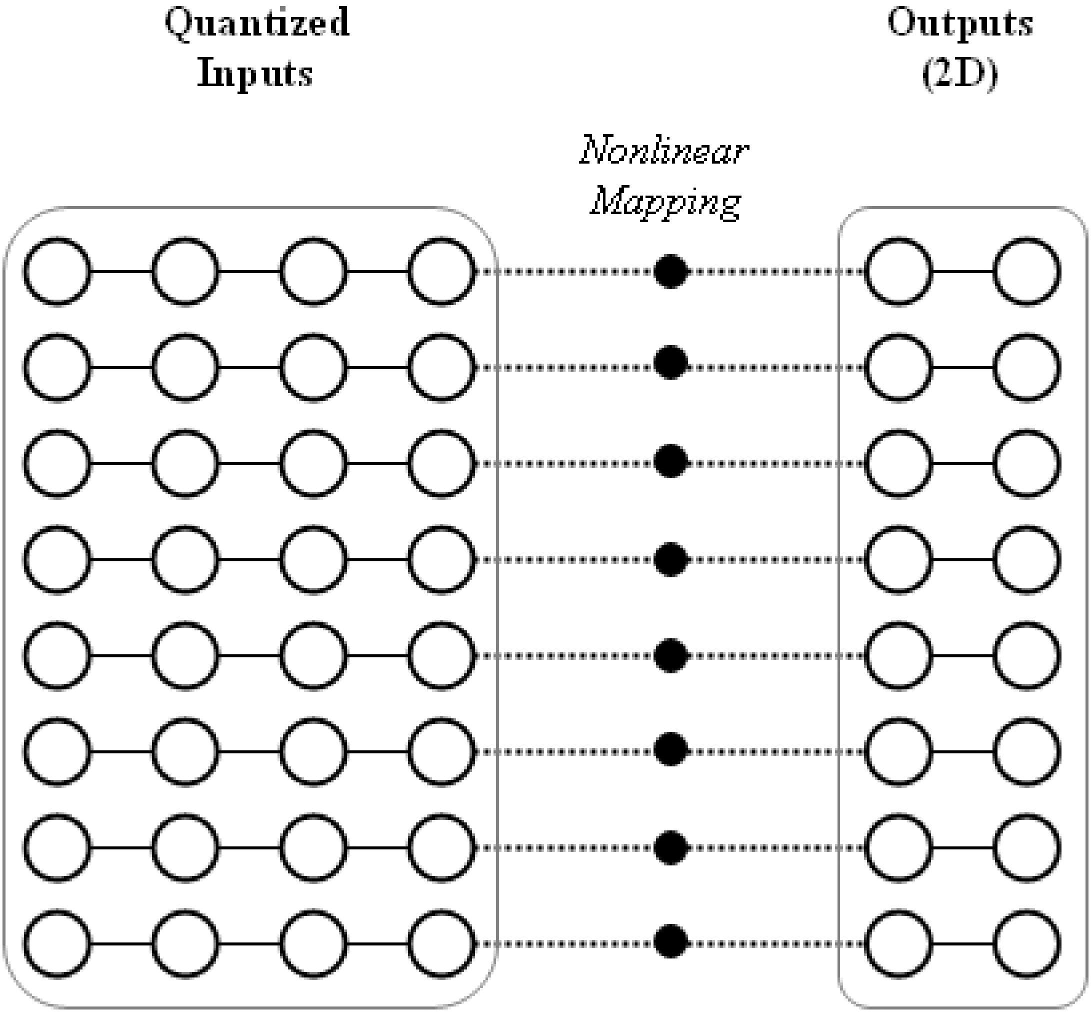


Figure 4

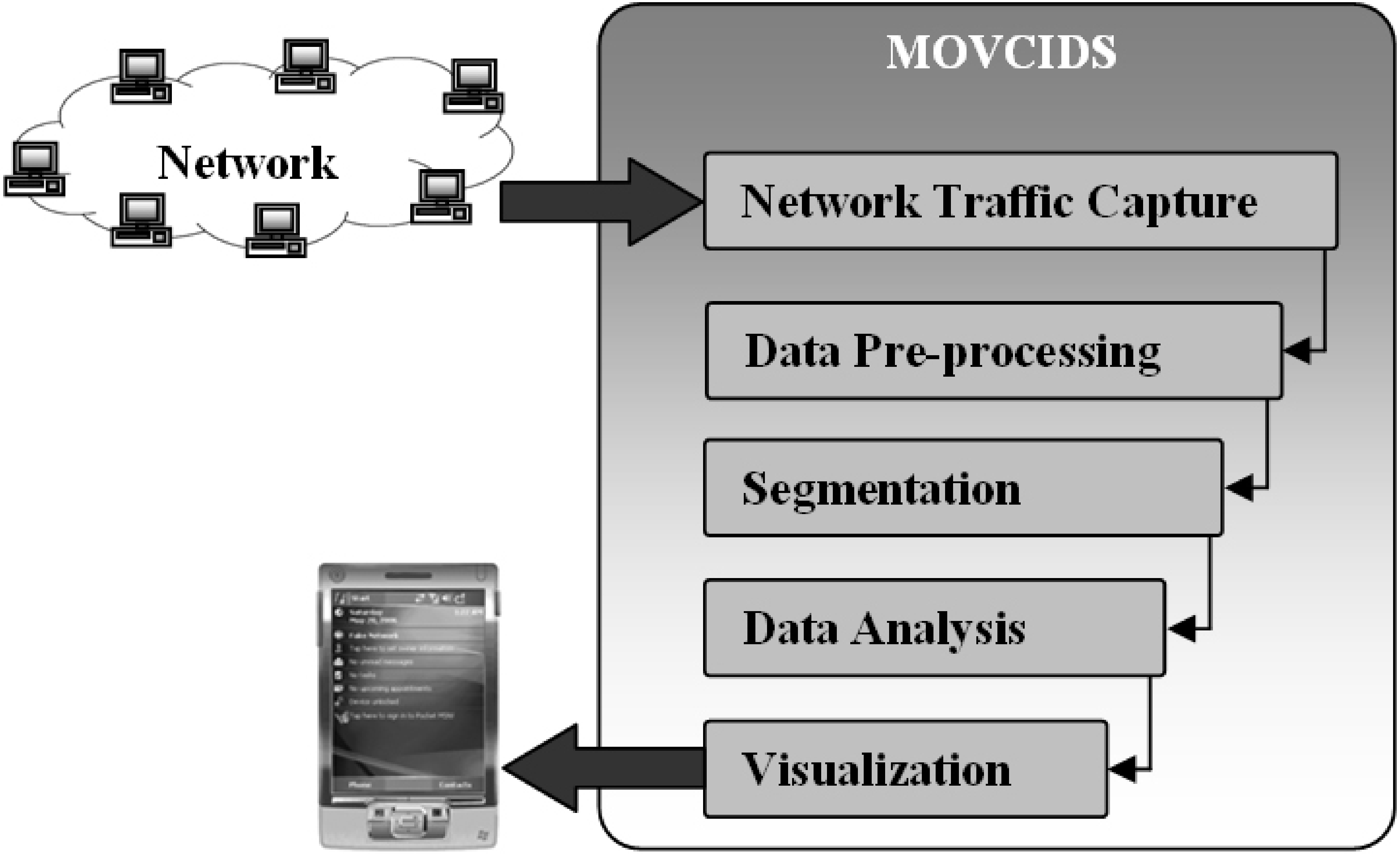

Network
MOVCIDS
Network Traffic Capture
Data Pre-processing
Segmentation
Data Analysis
Visualization

Figure 5

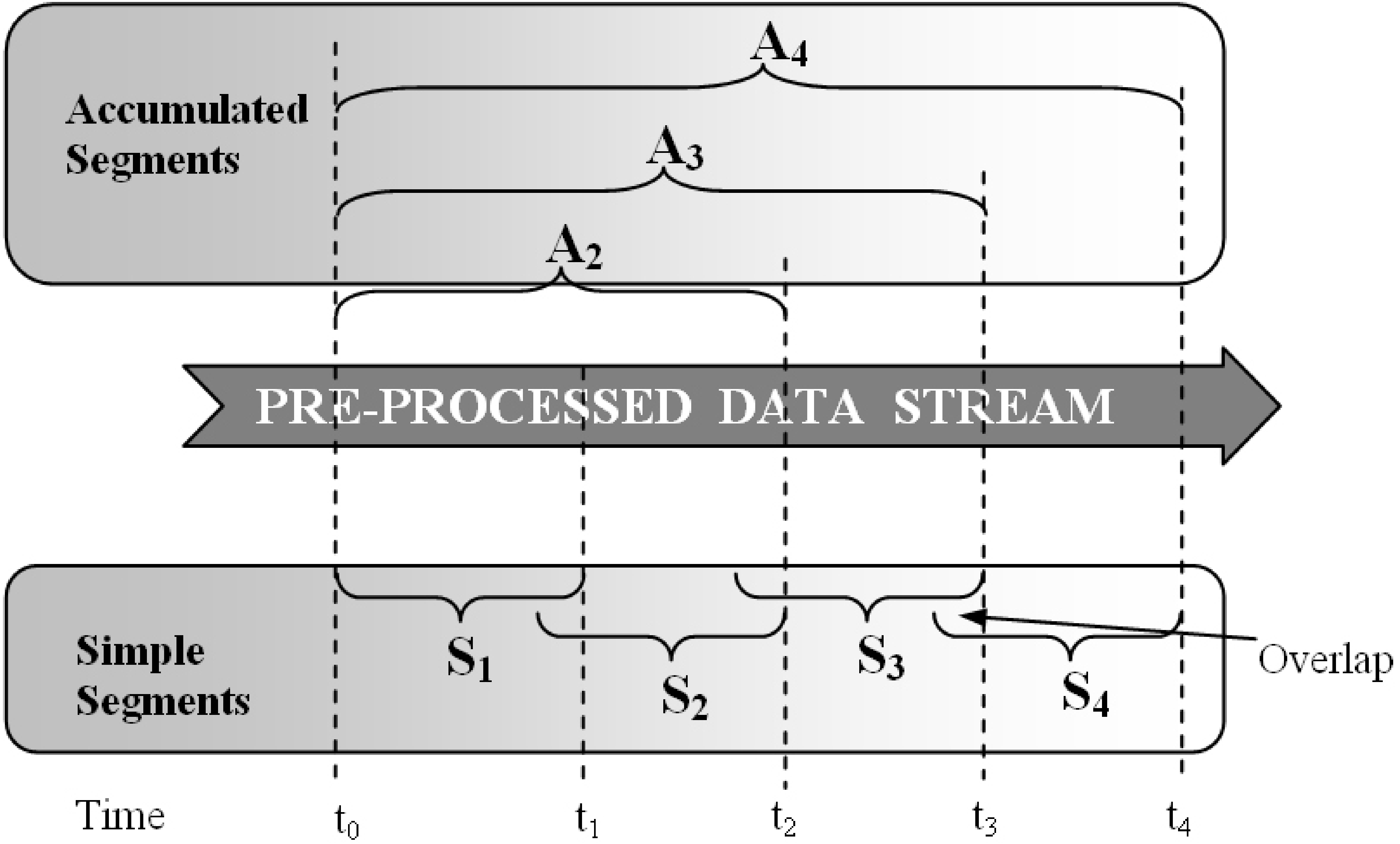

Accumulated Segments
A4
A3
A2
PRE-PROCESSED DATA STREAM
Simple Segments
S1
S2
S3
S4
Overlap
Time
t0
t1
t2
t3
t4

Figure 6

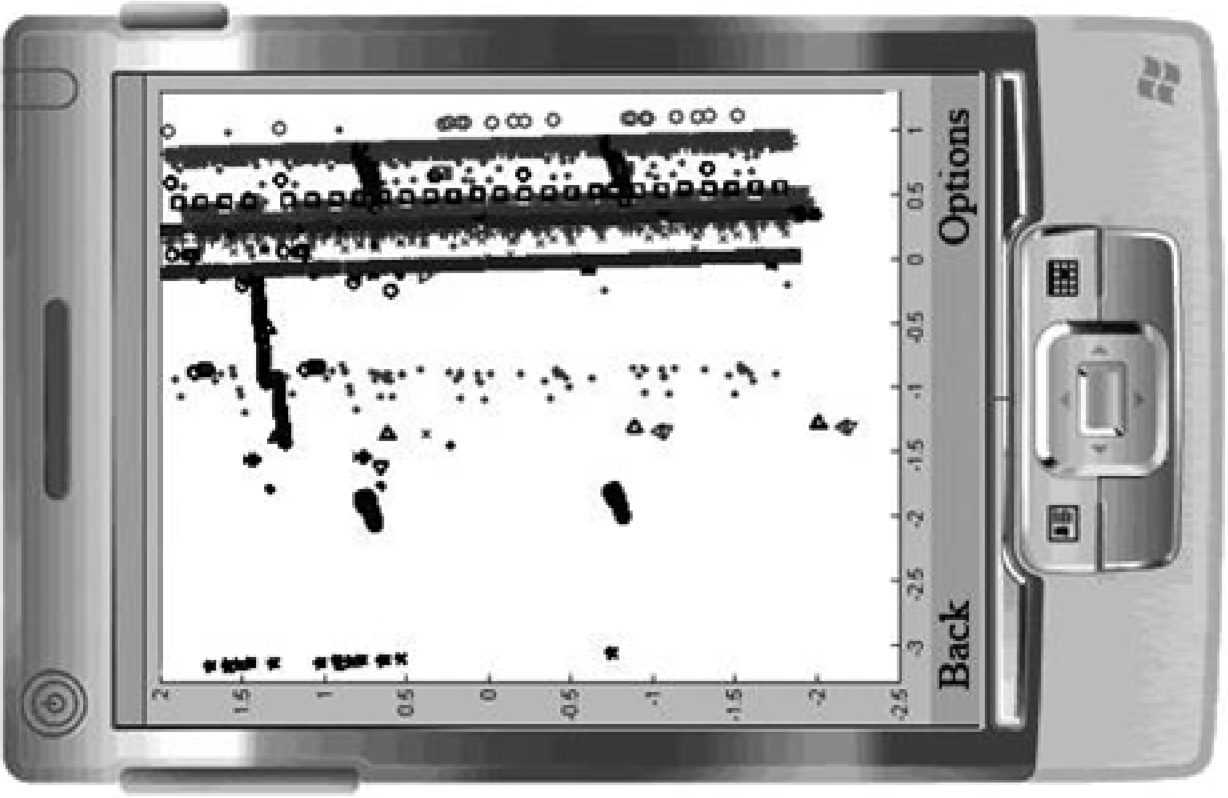


Figure 7a

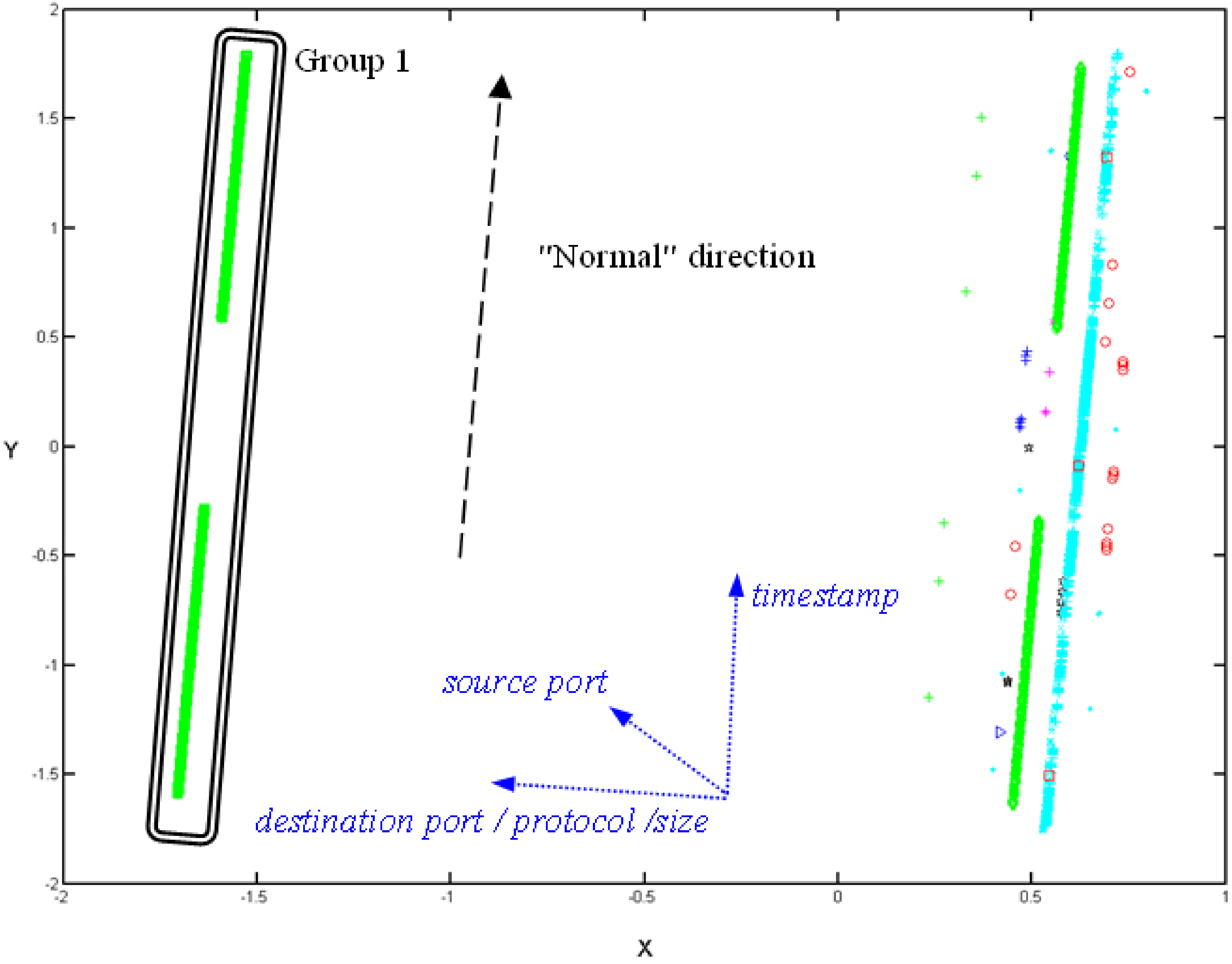

Group 1
"Normal" direction
timestamp
source port
destination port / protocol /size
Y
X
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1

Figure 7b

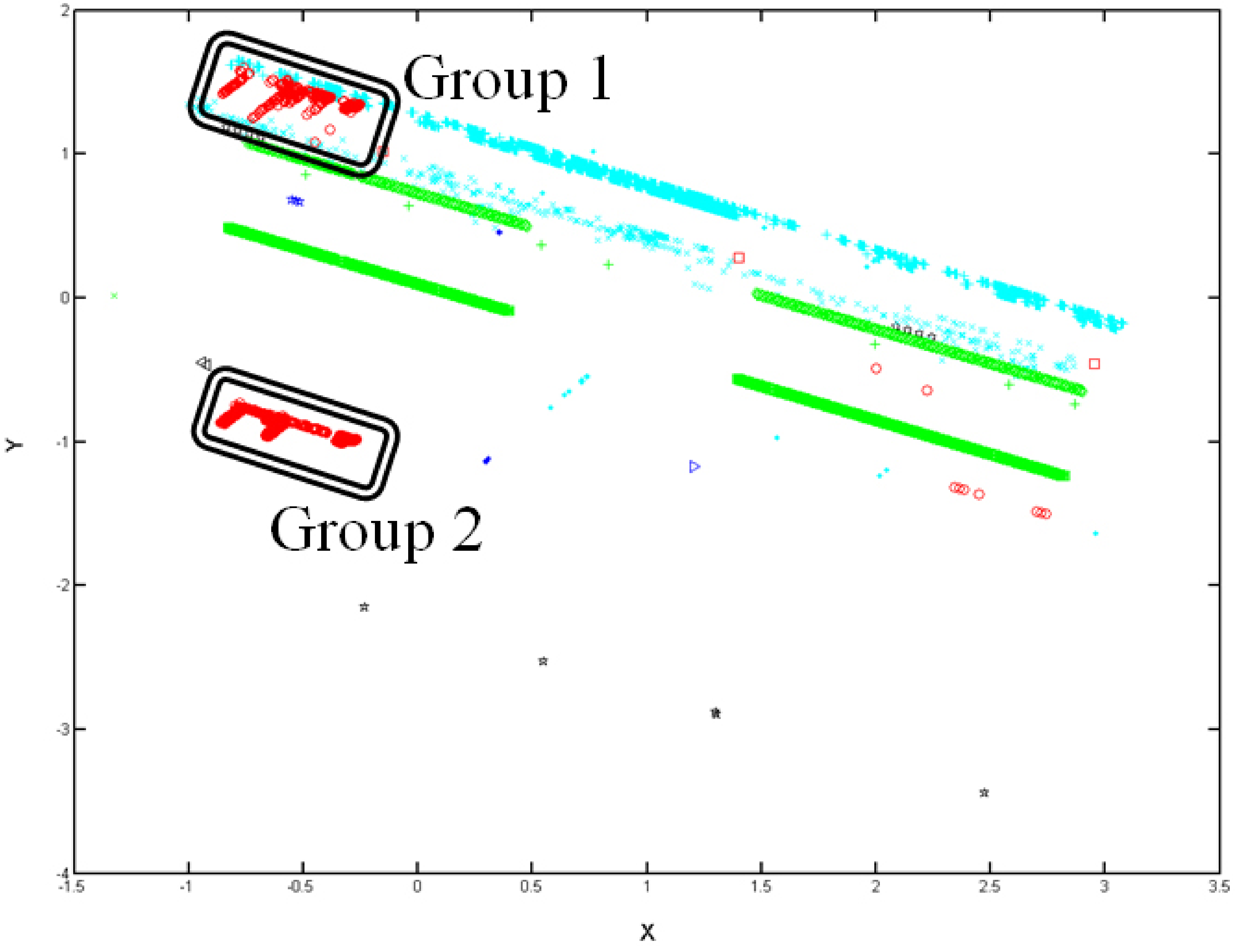


Figure 7c

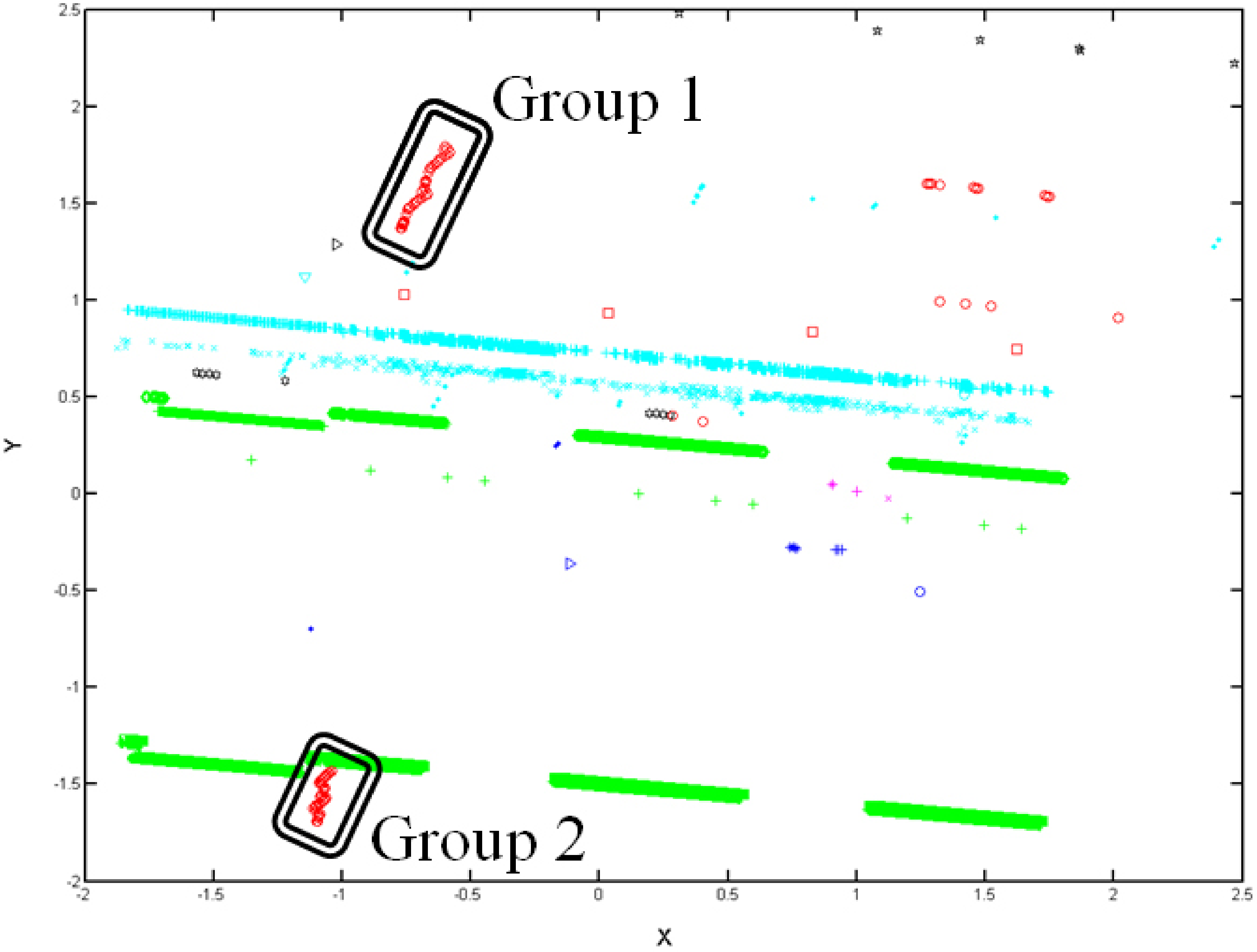


Figure 7d

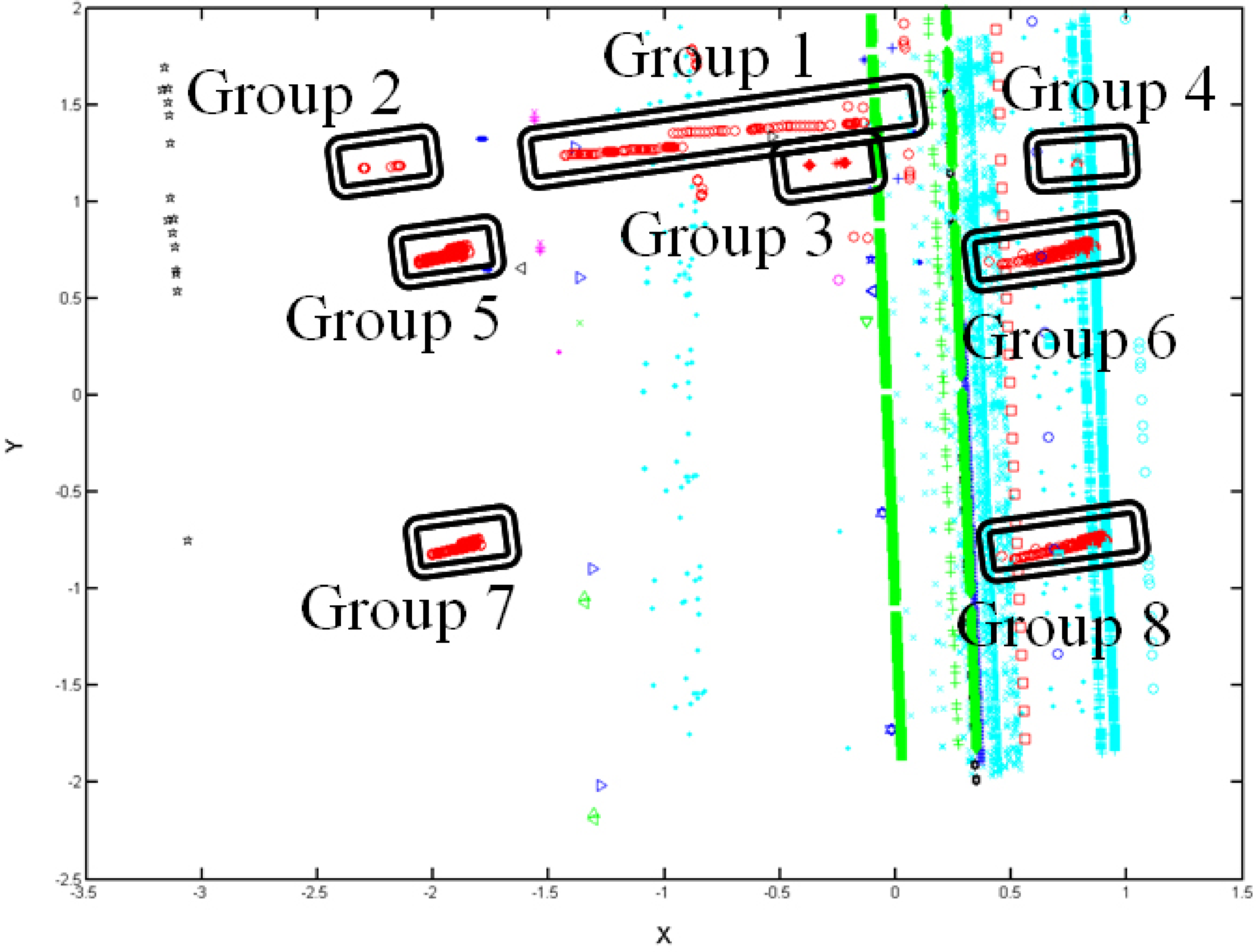

Group 1
Group 2
Group 3
Group 4
Group 5
Group 6
Group 7
Group 8
Y
X
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5

Figure 8a

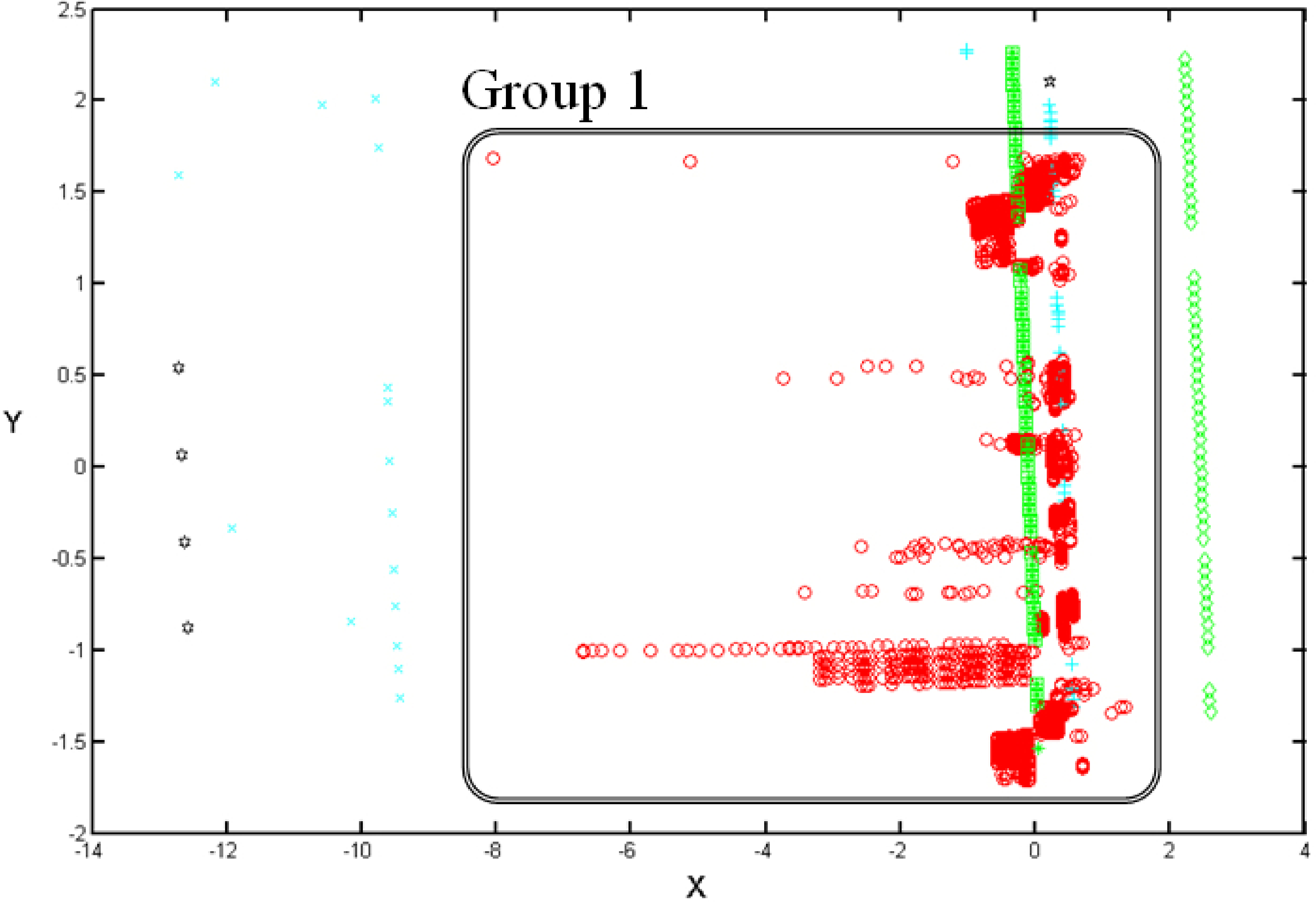


Figure 8b

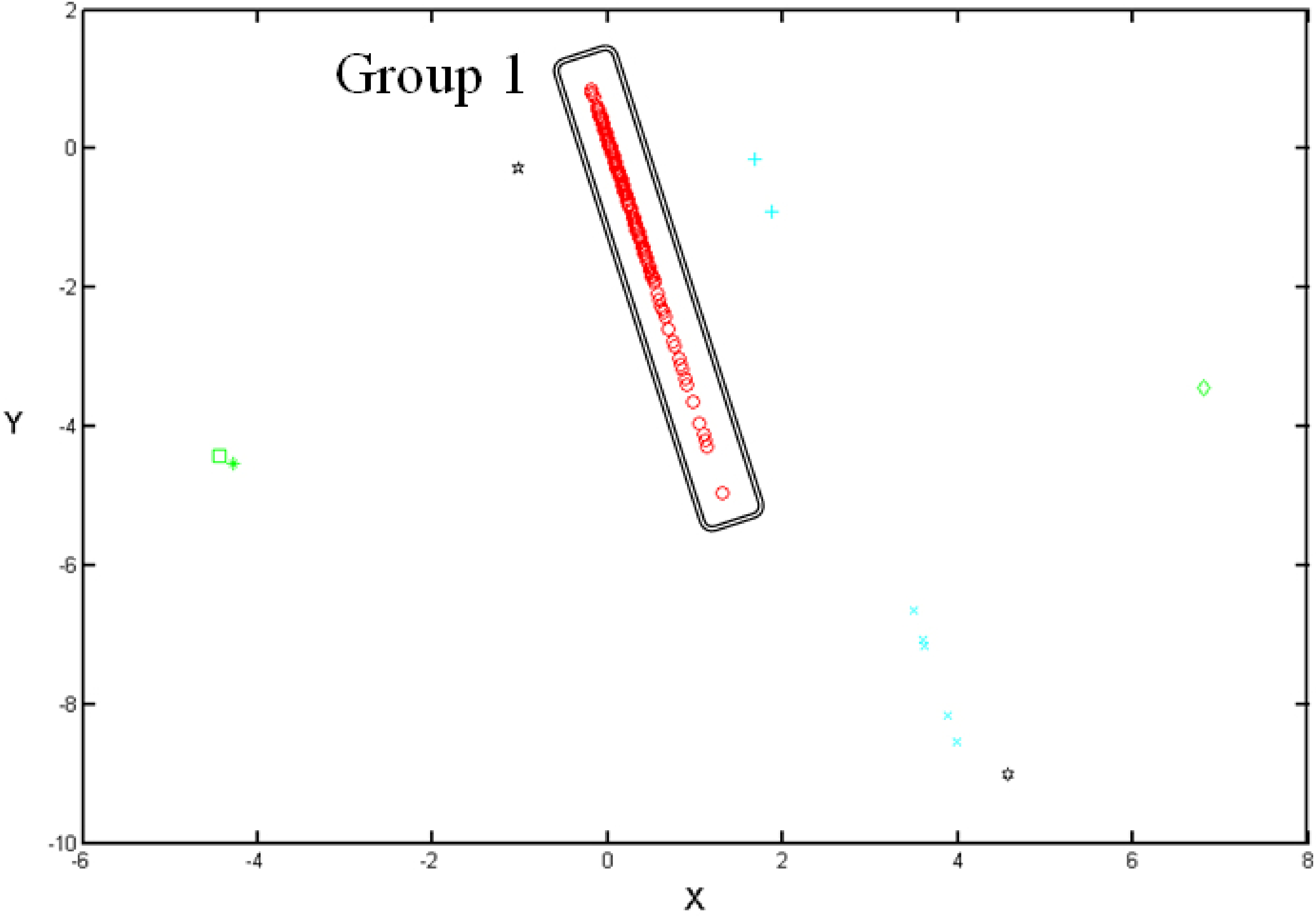
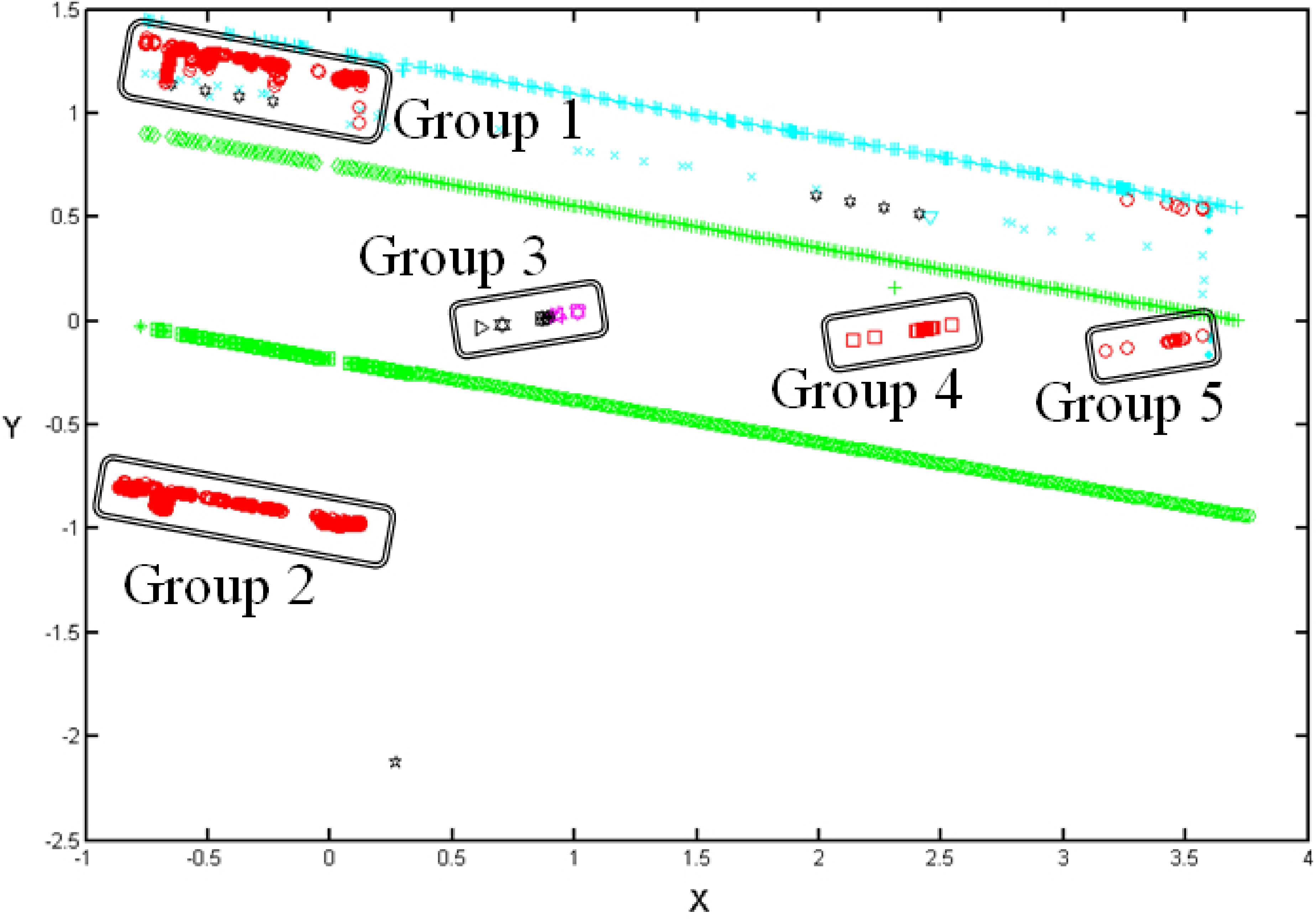


Figure 8c



Group 1
Group 2
Group 3
Group 4
Group 5
Y
X
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4

Figure 8d

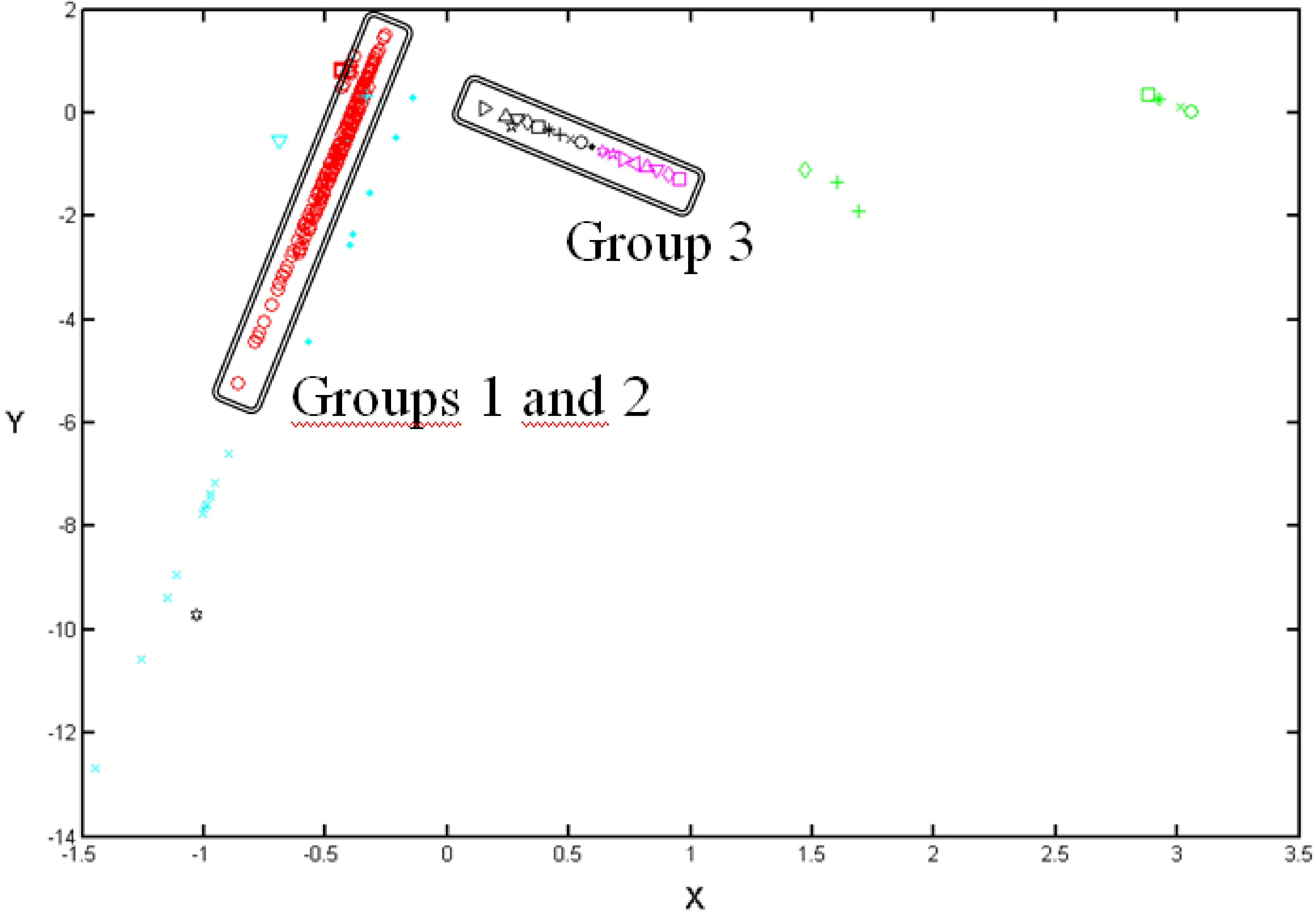


Figure 9

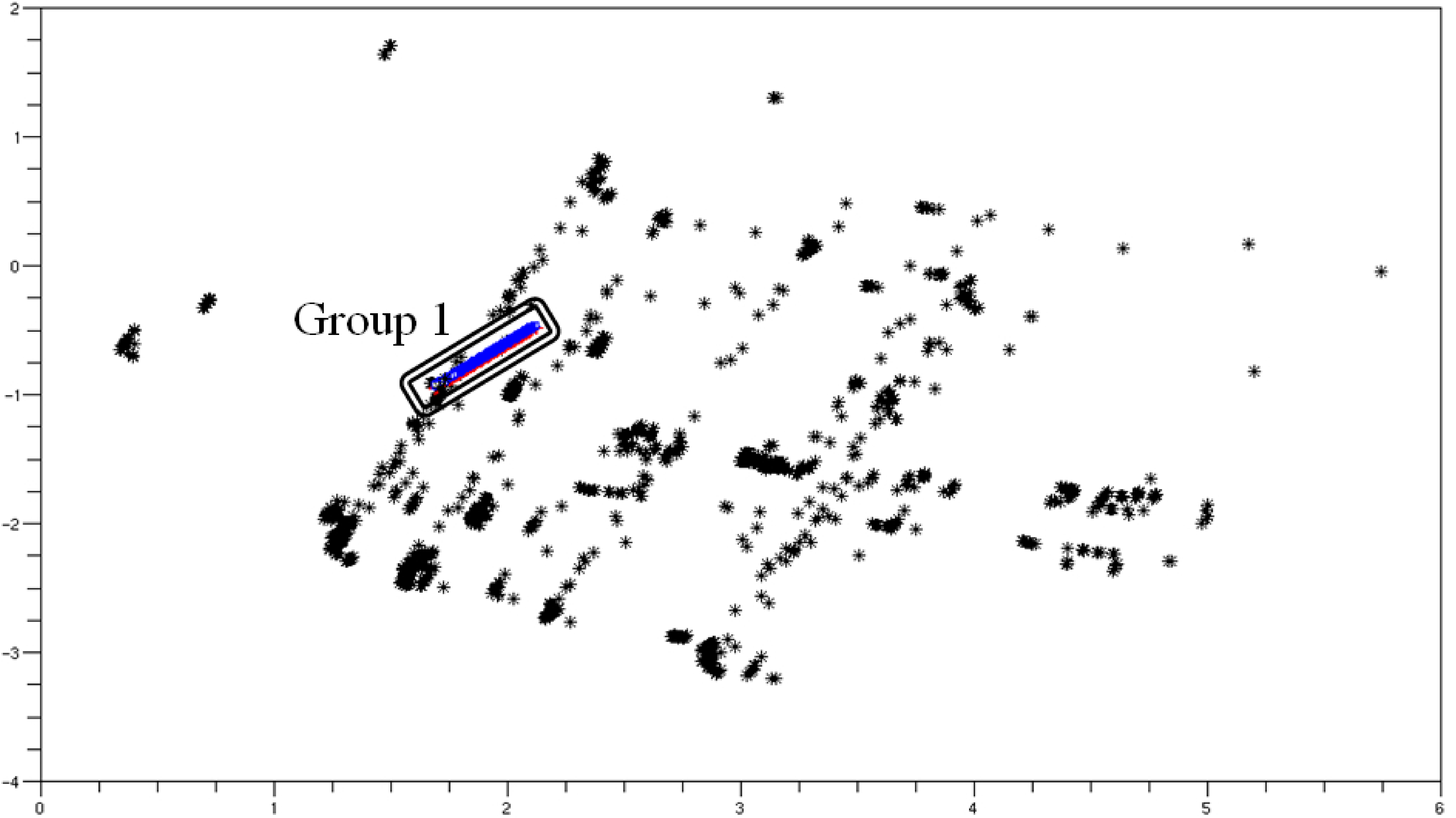


Figure 10a

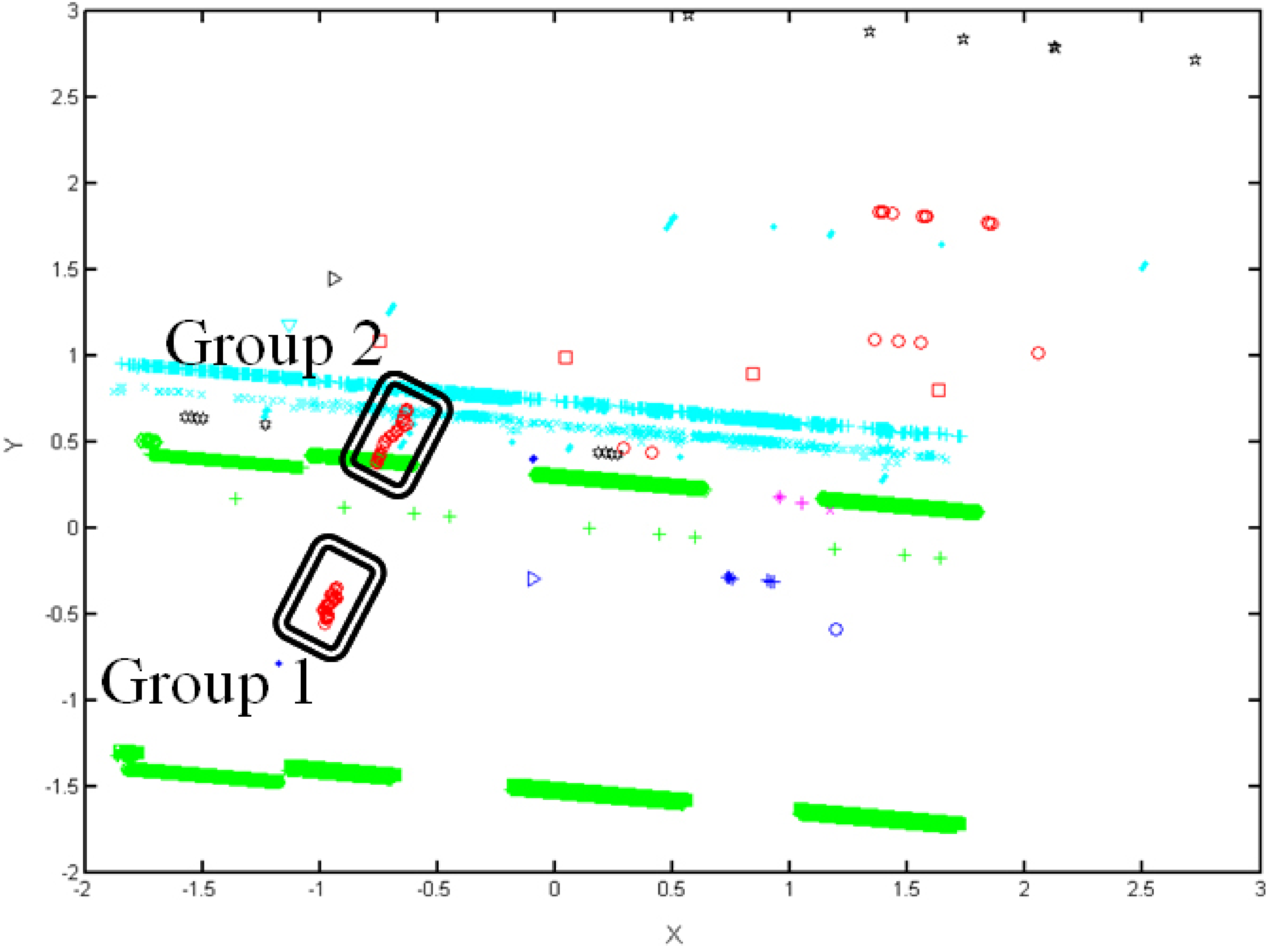


Figure 10b

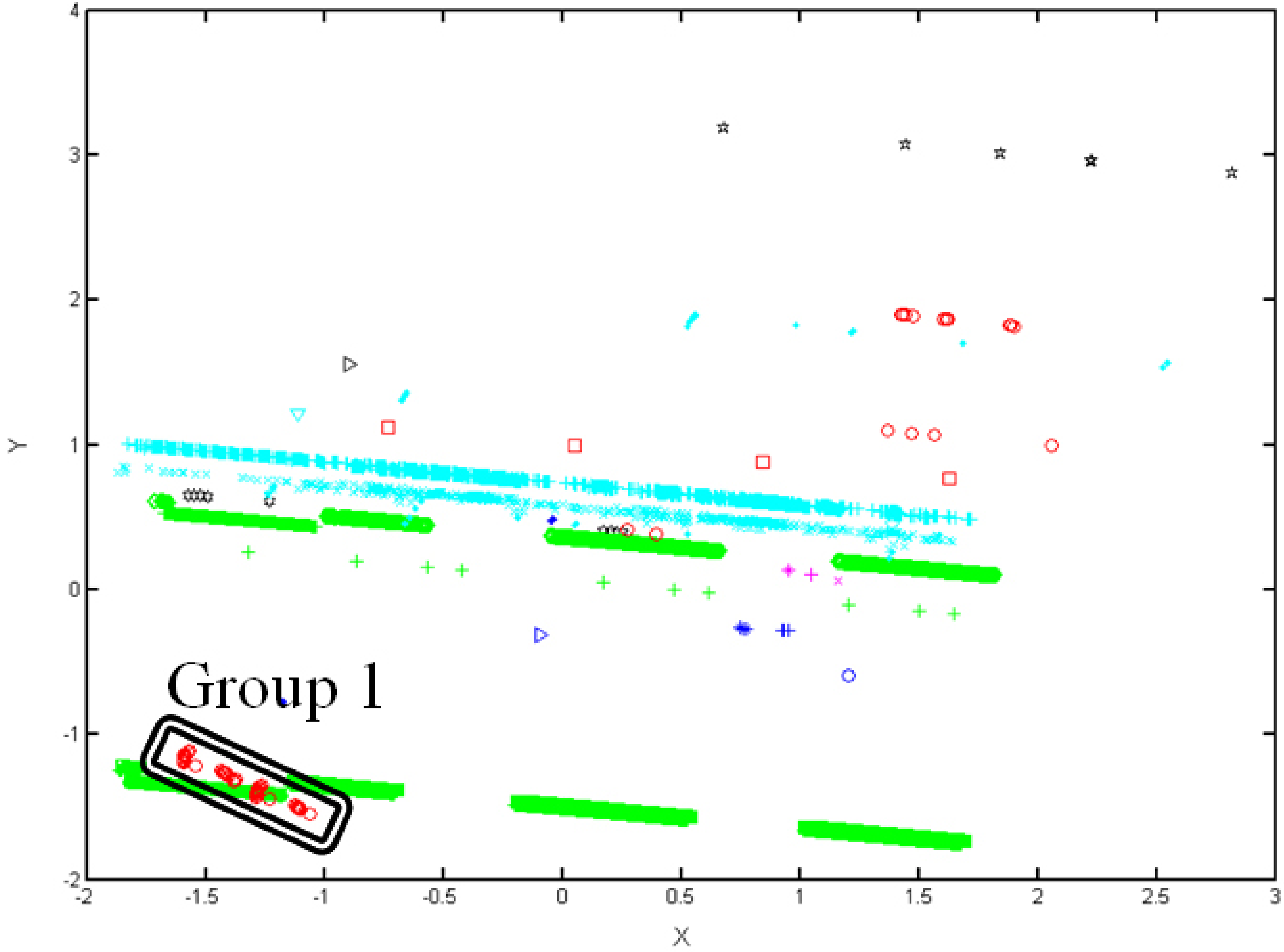


Figure 11a

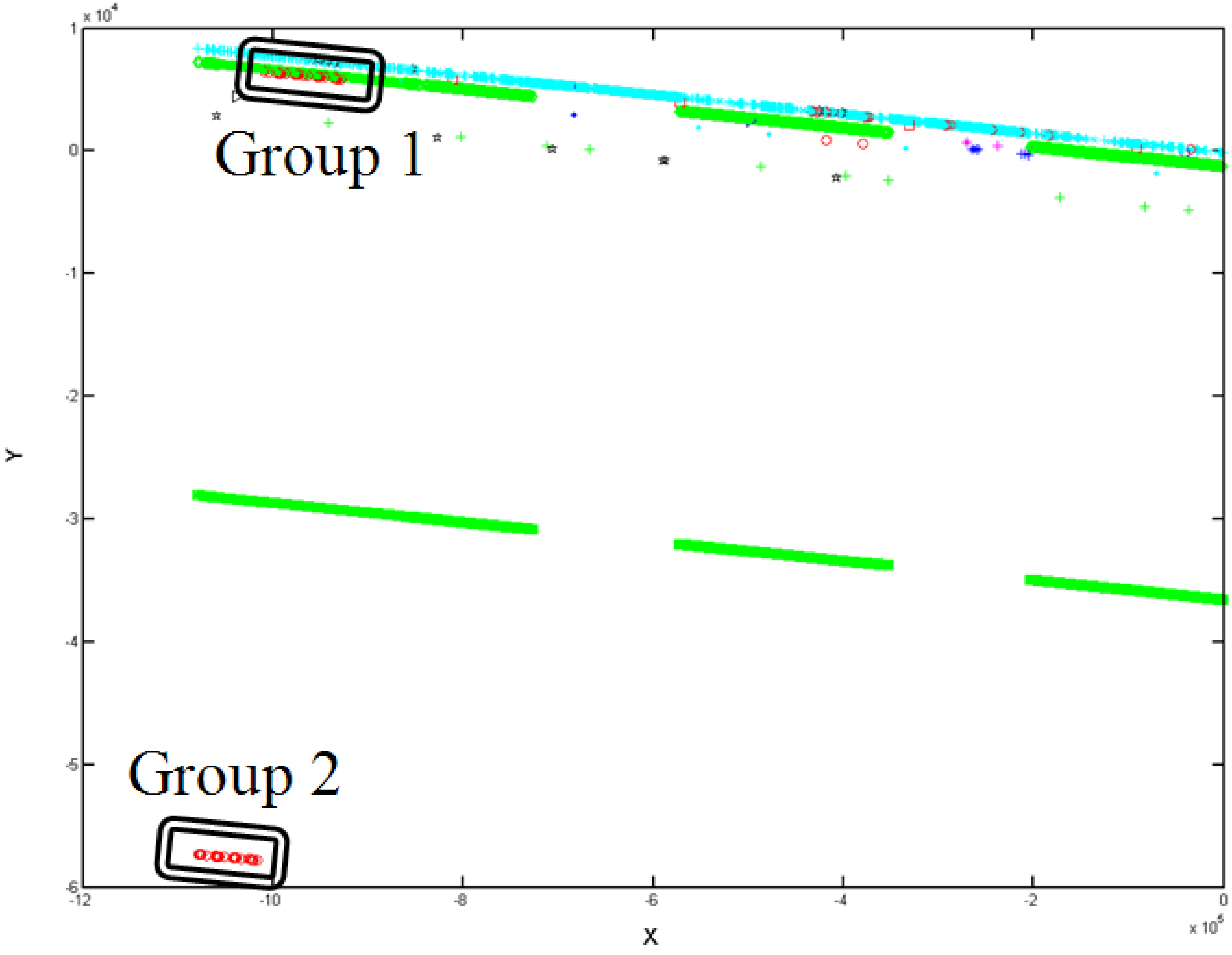


Figure 11b

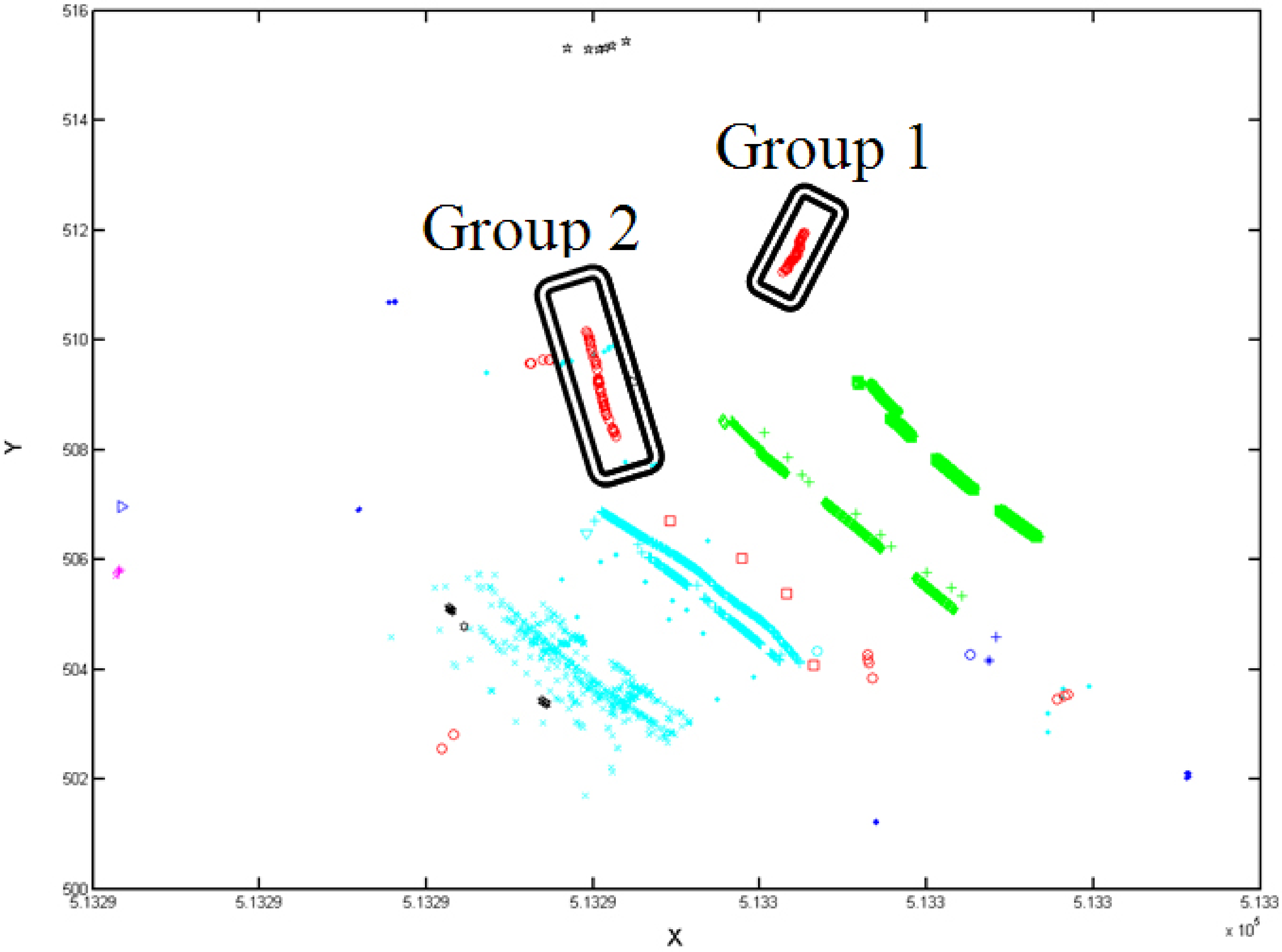

Group 1
Group 2
516
514
512
510
508
506
504
502
500
Y
5.1329
5.1329
5.1329
5.1329
5.133
5.133
5.133
5.133
X
x 10^5

Figure 11c

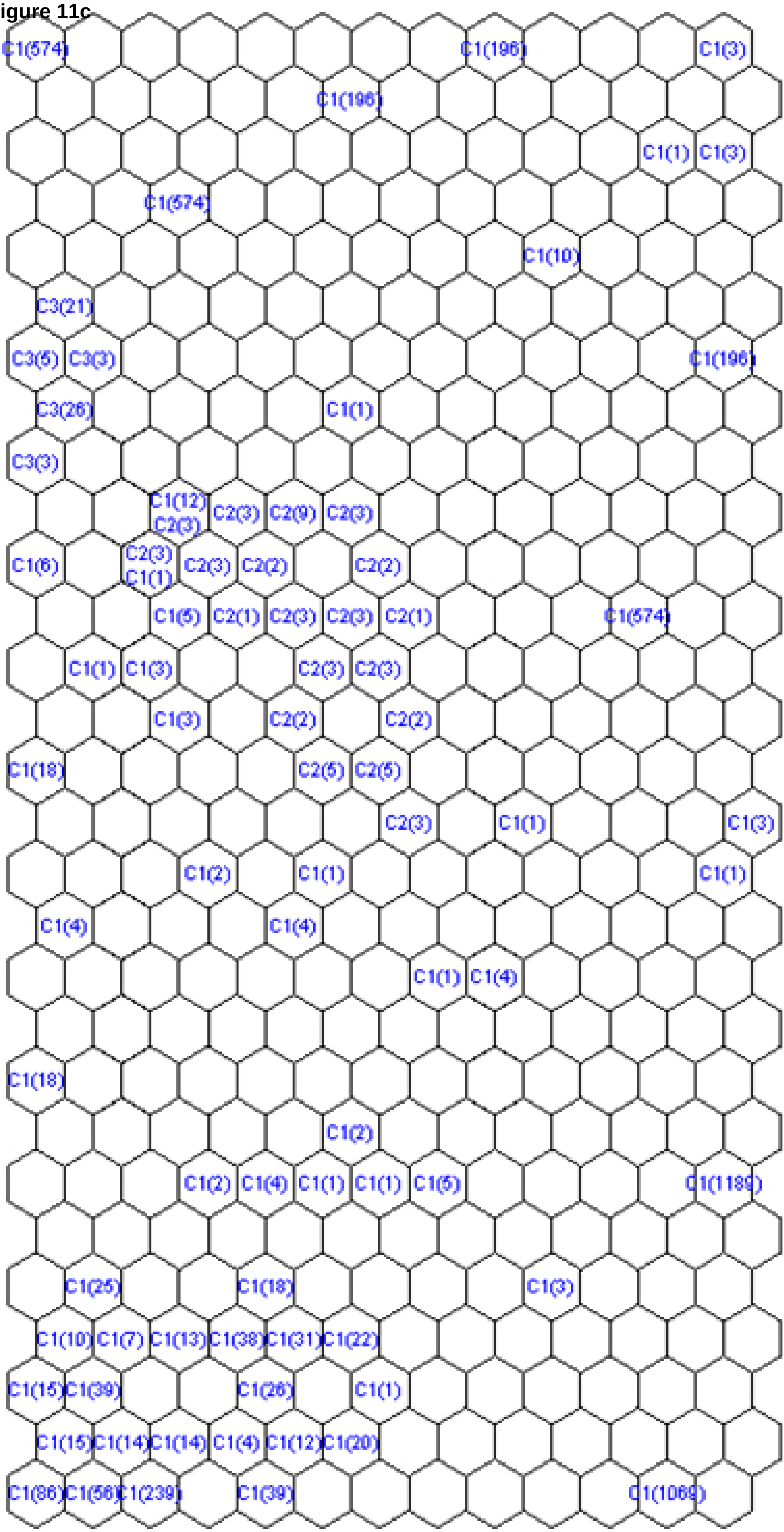
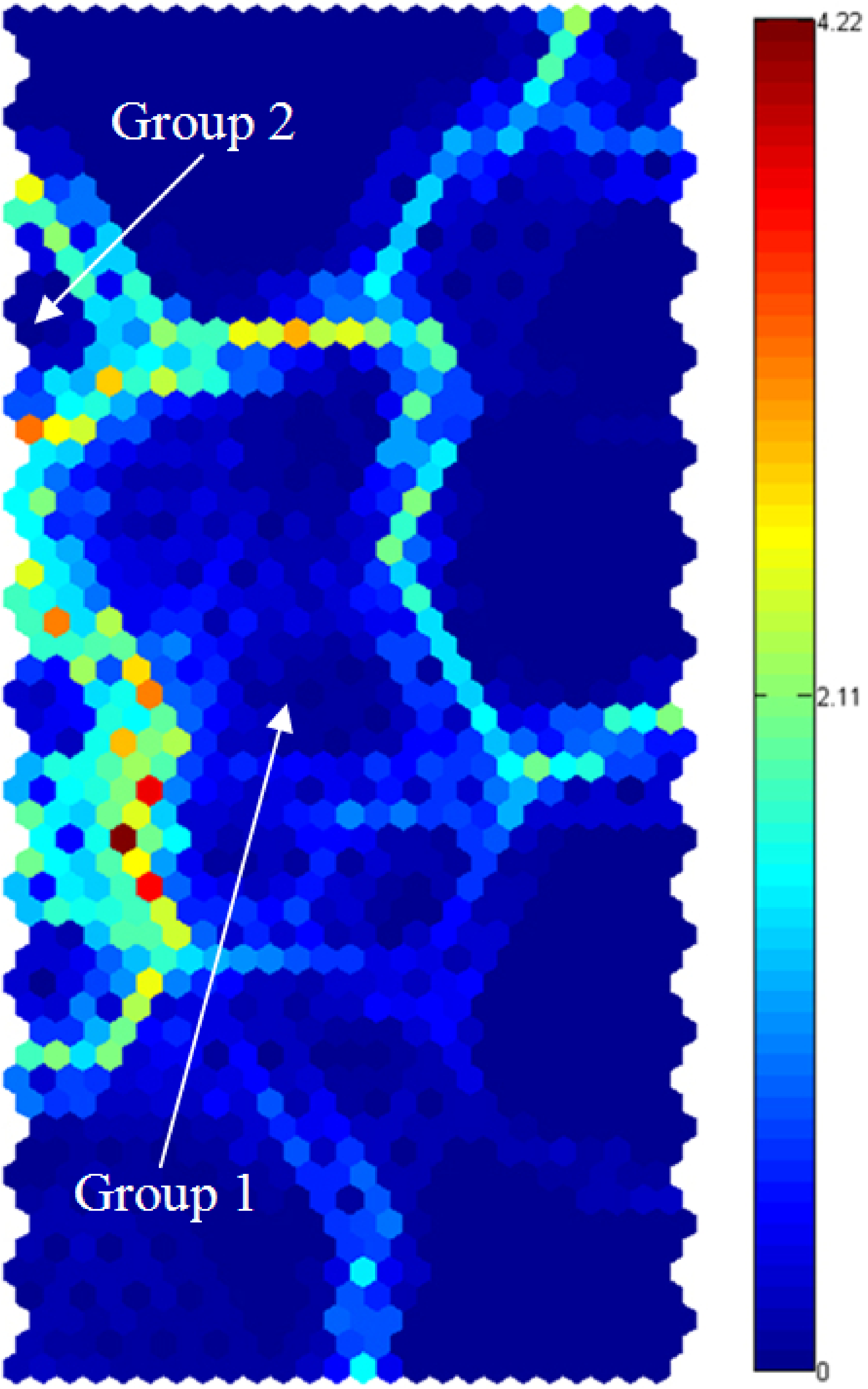

C1(574)
C1(196)
C1(3)
C1(196)
C1(1)
C1(3)
C1(574)
C1(10)
C3(21)
C3(5)
C3(3)
C1(196)
C3(26)
C1(1)
C3(3)
C1(12)
C2(3)
C2(3)
C2(9)
C2(3)
C1(6)
C2(3)
C1(1)
C2(3)
C2(2)
C2(2)
C1(5)
C2(1)
C2(3)
C2(3)
C2(1)
C1(574)
C1(1)
C1(3)
C2(3)
C2(3)
C1(3)
C2(2)
C2(2)
C1(18)
C2(5)
C2(5)
C2(3)
C1(1)
C1(3)
C1(2)
C1(1)
C1(1)
C1(4)
C1(4)
C1(1)
C1(4)
C1(18)
C1(2)
C1(2)
C1(4)
C1(1)
C1(1)
C1(5)
C1(1189)
C1(25)
C1(18)
C1(3)
C1(10)
C1(7)
C1(13)
C1(38)
C1(31)
C1(22)
C1(15)
C1(39)
C1(26)
C1(1)
C1(15)
C1(14)
C1(14)
C1(4)
C1(12)
C1(20)
C1(86)
C1(56)
C1(239)
C1(39)
C1(1069)

Figure 11d


Group 2
Group 1