University at Buffalo
The State University of New York

Data Mining for Intrusion Detection – Techniques, Applications and Systems

Jian Pei, Shambhu J. Upadhyaya
Faisal Farooq, Venugopal Govindaraju
State University of New York at Buffalo
{jianpei, shambhu, ffarooq2, govind}@cse.buffalo.edu

cse@buffalo

Outline

- Introduction
  - Intrusion: what and why?
  - Misuse detection and anomaly detection
  - Intrusion detection: bottom-line and challenges
- Data mining techniques for intrusion detection
  - Frequent pattern mining, classification, clustering, mining data streams
- Conclusions

What Are Intrusions?

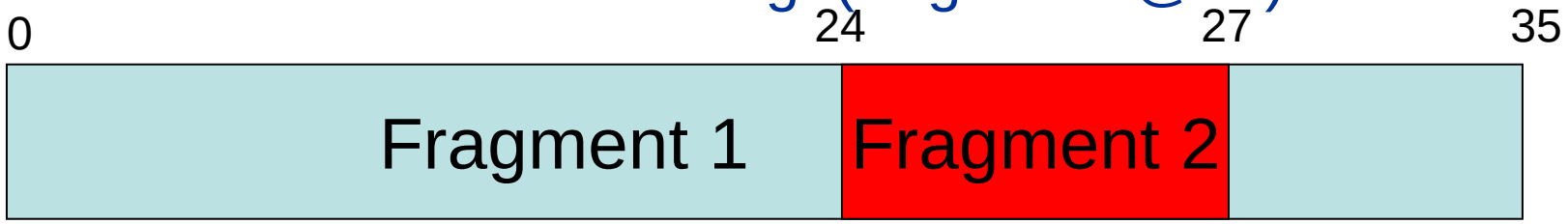
- Intrusions: any set of actions that threatens the integrity, availability, or confidentiality of a network resource
- Examples
  - Denial of service (DoS): attempts to starve a host of resources needed to function correctly
  - Scan: reconnaissance on the network or a particular host
  - Worms and viruses: replicating on other hosts
  - Compromises: obtain privileged access to a host by known vulnerabilities

Why Intrusions?

- Protocol abuse
  - The Teardrop program creates fragments with overlapping offset fields
  - The Smurf attack sends traffic to the broadcast address
- Holes in protocol implementations
  - Exceptional conditions that the implementers believed would never happen
  - E.g., some intrusion detection systems might not pick up an FIN scan

Example – Abusing Protocol

- Teardrop
  - attacker.com.139 > victim.org.139: udp 28 (frag 242:36@0+)
  - attacker.com > victim.org: (frag 242:4@24)



- Each fragment is legal
- When the fragments are reassembled at the destination host, may lead to crash, hang or reboot

Example – Holes in Implementation

- Most intrusion detection systems use SYN as signature to detect scans
- Some intrusion detection systems might not pick up an FIN scan
- FIN scan
  - nmap –sF victim.com: a stealthy FIN scan
  - attacker.com.38981 > victim.com.53: F 0:0(0) win 4096 (DF)

Intrusion Detection

- The process of monitoring and analyzing the events occurring in a computer and/or network system in order to detect signs of security problems
- Basic approaches: known pattern templates, threatening behavior templates, traffic analysis, statistical-anomaly detection and state-based detection
- Steps
  - Monitoring and analyzing traffic
  - Identifying abnormal activities
  - Assessing severity and raising alarm

Monitoring and Analyzing Traffic

- TCPdump and Windump
  - Provide insight into the traffic activity on a network
    - ftp://ftp.ee.lbl.gov/tcpdump.tar.Z
    - http://netgroupserv.polito.it/windump
- Ethereal
  - GUI to interpret all layers of the packet

Tradeoff: Functionality and Speed

- IDS performs more functions → slower in processing traffic → may have to drop packets
- Detect evasion or insertion attacks using host-based intrusion detection systems on resources
  - Host-based intrusion detection systems see the same packets that the hosts see
  - Application-level savvy is needed

Misuse vs. Anomaly Detection

- Misuse detection: Classification based on known intrusions
- Anomaly detection: Any significant deviations from the expected behavior are reported as possible attacks

Misuse Detection

- Human analysts investigate suspicious traffic
- Extract features of known intrusions
- Use pre-defined signatures to discover malicious packets
- Examples
  - Snort and Snort rules

Snort

- An open source free network intrusion detection system
  - Signature-based, uses a combination of rules and preprocessors
  - On many platforms, including UNIX and Windows
  - www.snort.org
- Preprocessors
  - IP defragmentation, port-scan detection, web traffic normalization, TCP stream reassembly, …
  - Can analyze streams, not only a single packet at a time

Snort Rules

- Two parts
  - Rule header: define who must be involved
  - Rule options: define what must be involved (action)

Rule header	Rule options
alert tcp !1.2.3.0/24 any -> 1.2.3.0/24 any	(flags: SF; msg:"SYN-FIN scan;)

- The rule triggers when an outsider attempts to make an internal TCP connection
- If both SYN and FIN are set, a message of "SYN-FIN scan" is reported with the alert

Application of Snort Rules

- A packet triggers the first rule that matches and does not examine the remainder
  - The ordering of rules is critical
- Each Snort rule inspects only one packet
- Use preprocessors such as IP defragmentation or TCP stream reassembly to handle a series of packets

Snort Rule Sets

- Snort comes with a very large set of rules
  - Not recommended that all rules used on installation
- New Snort rules are released as soon as hours after a new exploit is discovered
  - A new rule may not be a good rule
  - The attackers may change the signatures easily

Misuse Detection: Methods (1)

- Expert systems: use a set of rules to describe attacks
  - IDES, ComputerWatch, NIDX, P-BEST, ISOA
- Signature analysis: capture features of attacks in audit trail
  - Haystack, NetRanger, RealSecure, MuSig
- State-transition analysis: use state-transition diagrams
  - USTAT and NetSTAT

Misuse Detection: Methods (2)

- Data mining approaches
  - JAM and MADAM ID
- Other approaches
  - Colored Petri nets, e.g., IDIOT
  - Case-based reasoning, e.g., AUTOGUARD

Disadvantages of Misuse Detection

- Many false positives: prone to generating alerts when there is no problem in fact
  - Features are not specific enough
  - A packet is not examined in context with those that precede it or those that follow
- Cannot detect unknown intrusions
  - Rely on signatures extracted by human experts

Anomaly Detection

- Profiles: expected behavior
- Anomalies: significant deviations from the profiles
- Statistical methods: multivariate, temporal analysis
  - IDES, NIDES, EMERALD
- Expert systems
  - ComputerWatch, Wisdom & Sense
- Data mining
  - ADAM, IDDM, eBayes

Problems and Challenges

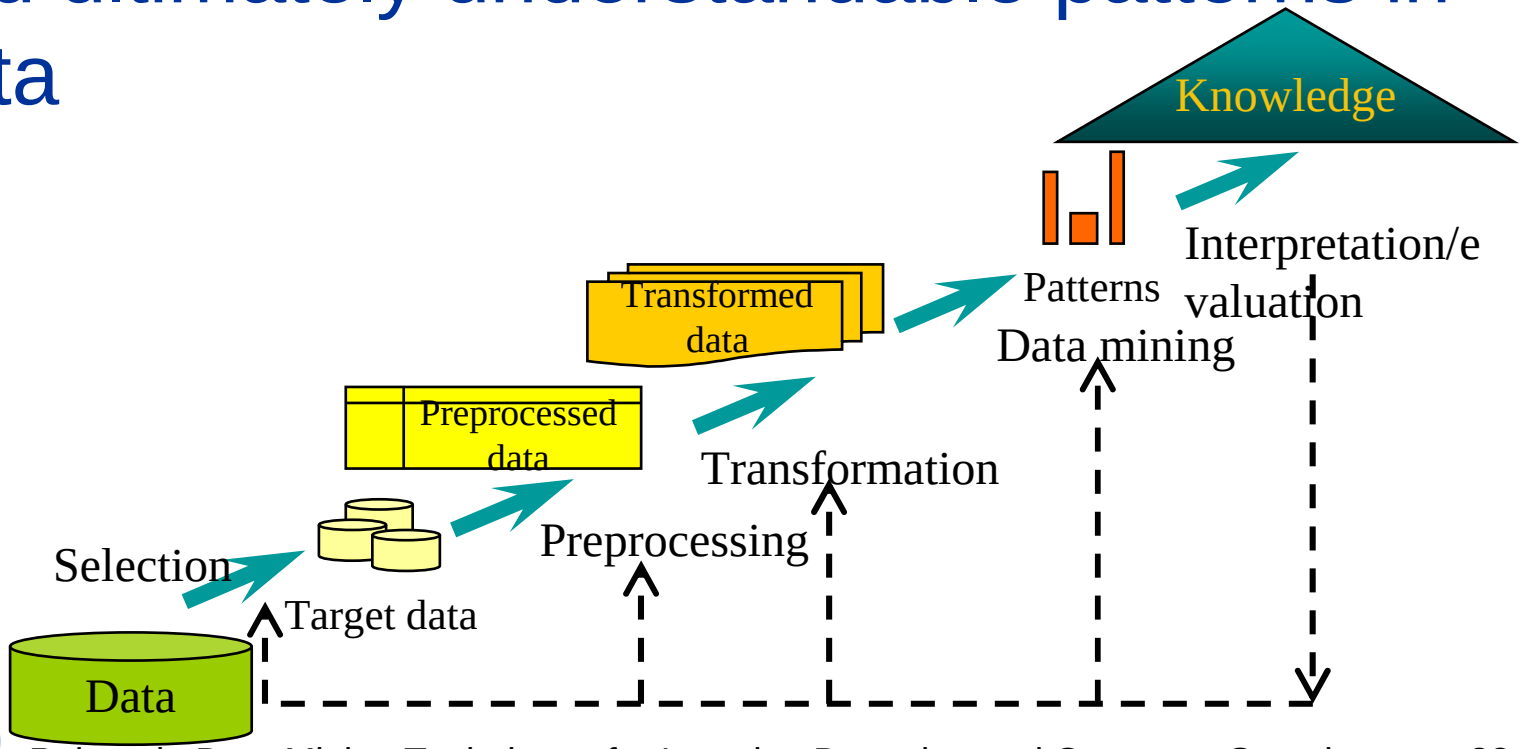
- How to detect known intrusions?
  - Find “patterns” of the known intrusions
  - Then, how to detect as accurate as possible?
- How to detect unknown intrusions?
  - Find “unusual patterns” of possible intrusions
  - Then, how to detect as accurate as possible?
- Efficiency and scalability concerns: how to detect in a large and fast network?

Outline

- Introduction
  - Intrusion: what and why?
  - Intrusion detection: bottom-line and challenges
- Data mining techniques for intrusion detection
  - Frequent pattern mining, classification, clustering, mining data streams
- Conclusions

What Is Data Mining?

- Data mining is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data



Why Can Data Mining Help?

- Learn from traffic data
  - Supervised learning: learn precise models from past intrusions
  - Unsupervised learning: identify suspicious activities
- Maintain models on dynamic data

Data Mining for Intrusion Detection: Techniques and Applications

- Frequent pattern mining
- Classification
- Clustering
- Mining data streams

What Are Frequent Patterns?

- Patterns (set of items, sequence, etc.) that occur frequently in a database
- Mining Frequent patterns – finding regularities
  - What products were often purchased together?
- Frequent patterns for intrusion detection
  - What are the frequent features in abnormal/malicious packets?

Basics

- Itemset: a set of items
  - E.g., acm={a, c, m}
- Support of itemsets
  - Sup(acm)=3
- Given min_sup=3, acm is a frequent pattern
- Frequent pattern mining: find all frequent patterns in a database

Transaction database TDB

TID	Items bought
100	f, a, c, d, g, l, m, p
200	a, b, c, f, l, m, o
300	b, f, h, j, o
400	b, c, k, s, p
500	a, f, c, e, l, p, m, n

From Frequent Patterns to Rules

- Association rule X → Y
  - Sup(X) = 10%
  - Sup(XY) = 8%
  - Sup(X→Y)=sup(XY)=8%
  - Confidence(X→Y)=sup(XY)/sup(X)=80%
- Strong and confident association rules
  - Strong – high support
  - Confident – high confidence

Extensions of Association Rules

- Boolean vs. quantitative associations
  - buys(x, “SQLServer”) ^ buys(x, “DMBook”) → buys(x, “DM Software”) [0.2%, 60%]
  - age(x, “30..39”) ^ income(x, “42..48K”) → buys(x, “PC”) [1%, 75%]
- Single dimension vs. multiple dimensional associations
- Single level vs. multiple-level analysis
  - What brands of diapers are associated with diapers?

Extensions & Applications

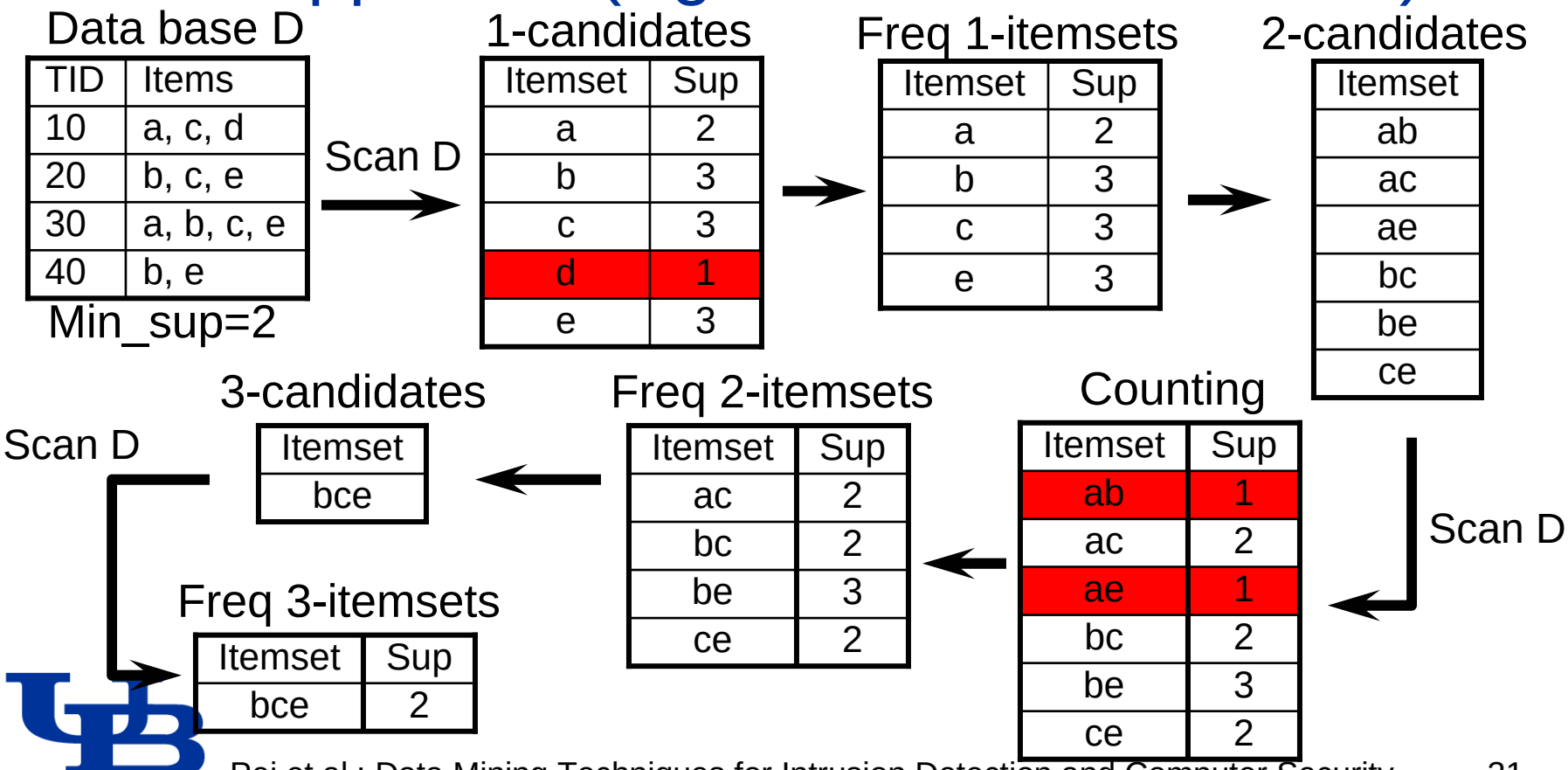
- Correlation, causality analysis & mining interesting rules
- Non-redundant frequent patterns
  - Maxpatterns and frequent closed itemsets
- Constraint-based mining

Apriori: Candidate Generation-and-test

- Any subset of a frequent itemset must be also frequent — an anti-monotone property
  - A transaction containing {beer, diaper, nuts} also contains {beer, diaper}
  - {beer, diaper, nuts} is frequent → {beer, diaper} must also be frequent
- No superset of any infrequent itemset should be generated or tested
  - Many item combinations can be pruned

Apriori Algorithm

- A level-wise, candidate-generation-and-test approach (Agrawal & Srikant 1994)



Sequence Databases and Sequential Pattern Analysis

- (Temporal) order is important in many situations
  - Time-series databases and sequence databases
  - Frequent patterns → (frequent) sequential patterns
- Applications of sequential pattern mining
  - First buy computer, then CD-ROM, and then digital camera, within 3 months.
  - Medical treatment, natural disasters prediction, DNA sequences and gene structures
- Sequential patterns for intrusion detection
  - Capture the signatures for attacks in a series of packets

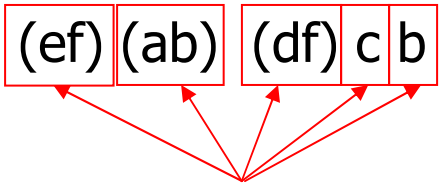
Sequential Pattern Mining

- Given a set of sequences, find the complete set of frequent subsequences

A *sequence database*

SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

A *sequence* : <(ef)(ab)(df)cb>



An element may contain a set of items. Items within an element are unordered and we list them alphabetically.

<a(bc)dc> is a *subsequence* of <a(abc)(ac)d(cf)>

Given *support threshold* *min_sup* =2, <(ab)c> is a *sequential pattern*

Apriori Property in Sequences

- Apriori property in sequential patterns
  - If a sequence S is infrequent, then none of the super-sequences of S is frequent
  - E.g, <hb> is infrequent → so do <hab> and <(ah)b>

Given *support threshold* *min_sup* =2

Seq-id	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

The GSP Mining Process

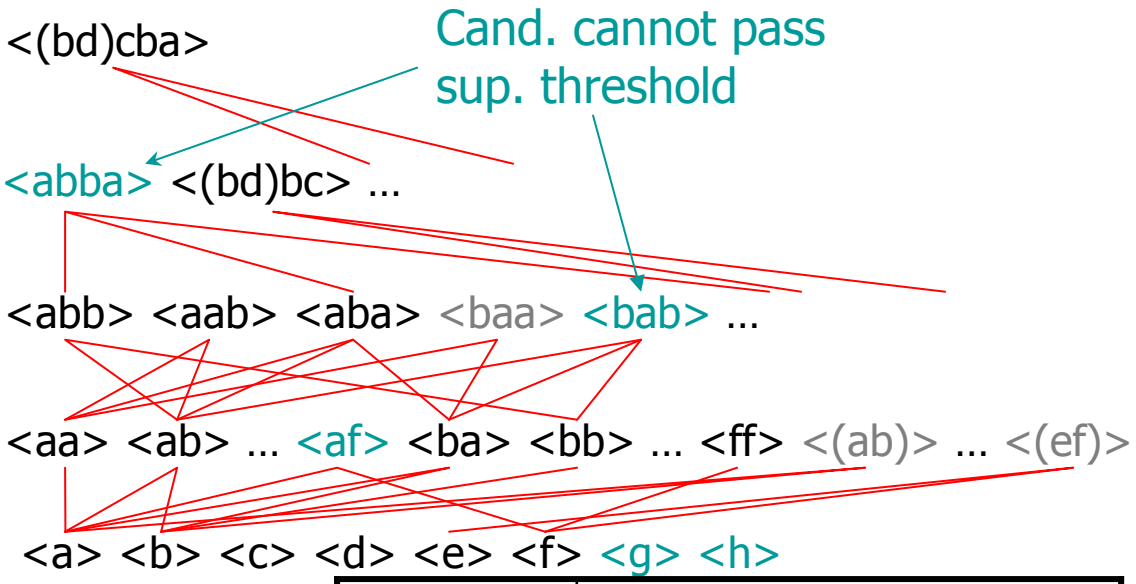
5th scan: 1 cand. 1 length-5 seq. pat.

4th scan: 8 cand. 6 length-4 seq. pat.

3rd scan: 46 cand. 19 length-3 seq. pat. 20 cand. not in DB at all

2nd scan: 51 cand. 19 length-2 seq. pat. 10 cand. not in DB at all

1st scan: 8 cand. 6 length-1 seq. pat.



min_sup =2

Seq-id	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Frequent Pattern Mining in ADAM

- "Detecting intrusions by data mining"
  - Barbara et al. @ George Mason University
- Phase I: mine a repository of normal frequent itemsets for attack-free data
- Phase II: find frequent itemsets in the current sliding window and compare the patterns to the normal profile
- Use a classifier to reduce false positives

Frequent Pattern Mining in MINDS

- MINDS: a IDS using data mining techniques
  - University of Minnesota
- Summarizing attacks using association rules
  - {Src IP=206.163.27.95, Dest Port=139, Bytes$\in$[150, 200)} → {ATTACK}

Specification-based Anomaly Detection

- Sekar et al. CCS'02
- Using state-machine to specify network protocols – the pattern templates
- Learning statistical properties
  - How frequently a transition is taken, or the commonly encountered values of state variables on a transition?
  - Use distribution histogram instead of average
- If the statistics during the detection phase differ from the profile substantially, an anomaly
  - Use thresholds

Patterns About Alerts

- Ning et al. CCS'02
- Find correlated alerts – the frequent patterns of alerts
  - Attack scenarios – the logical connections between alerts
  - A hyper-alerts correlation graph approach
- Use the correlation of intrusion alerts to identify high level attacks

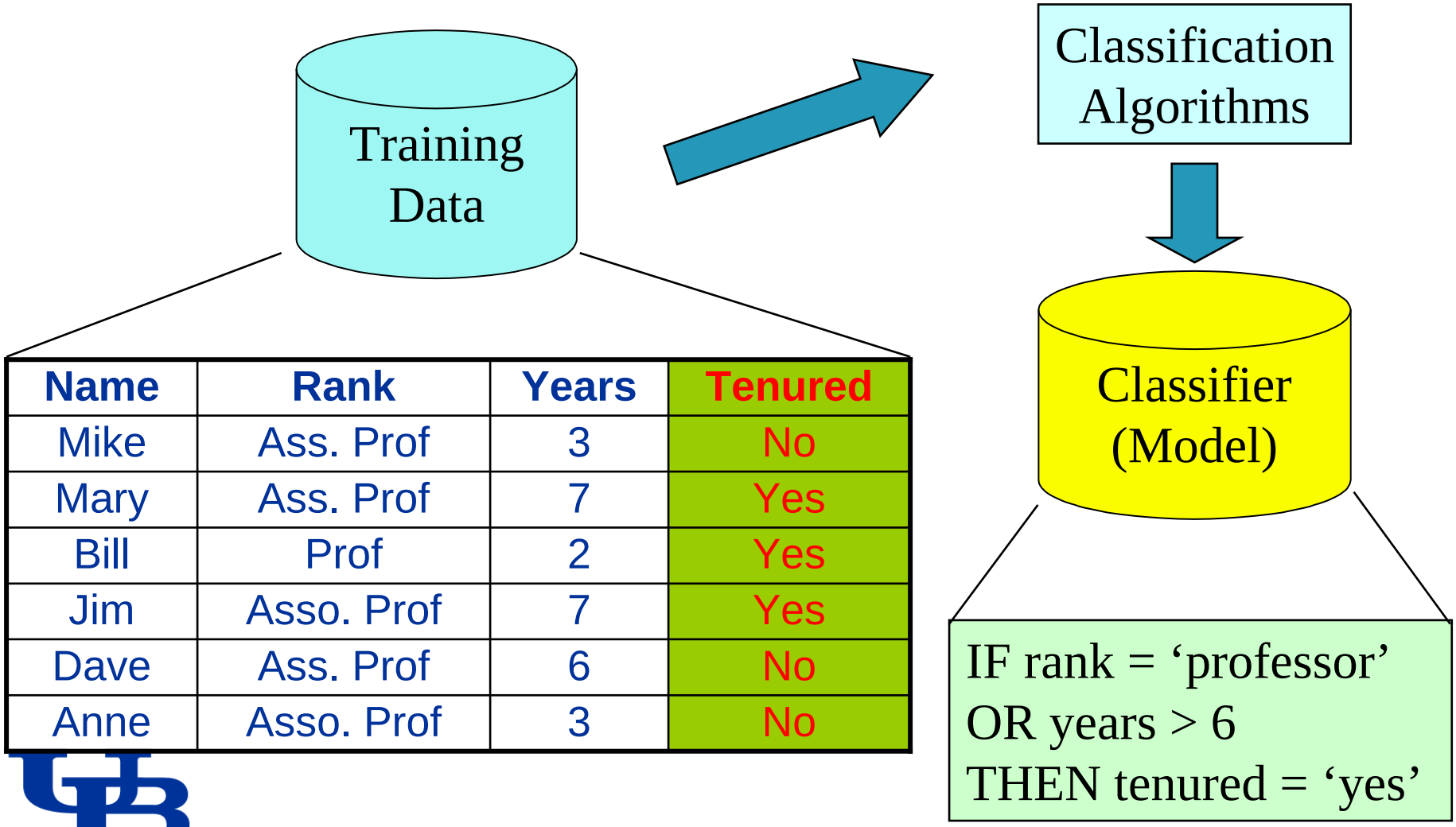
Classification and Prediction

- Classification: predict categorical class labels
  - Build a model for a set of classes/concepts
  - Classify bank loan applications (safe/risky)
- Prediction: model continuous-valued functions
  - Predict the economic growth in 2004

Classification: A 2-step Process

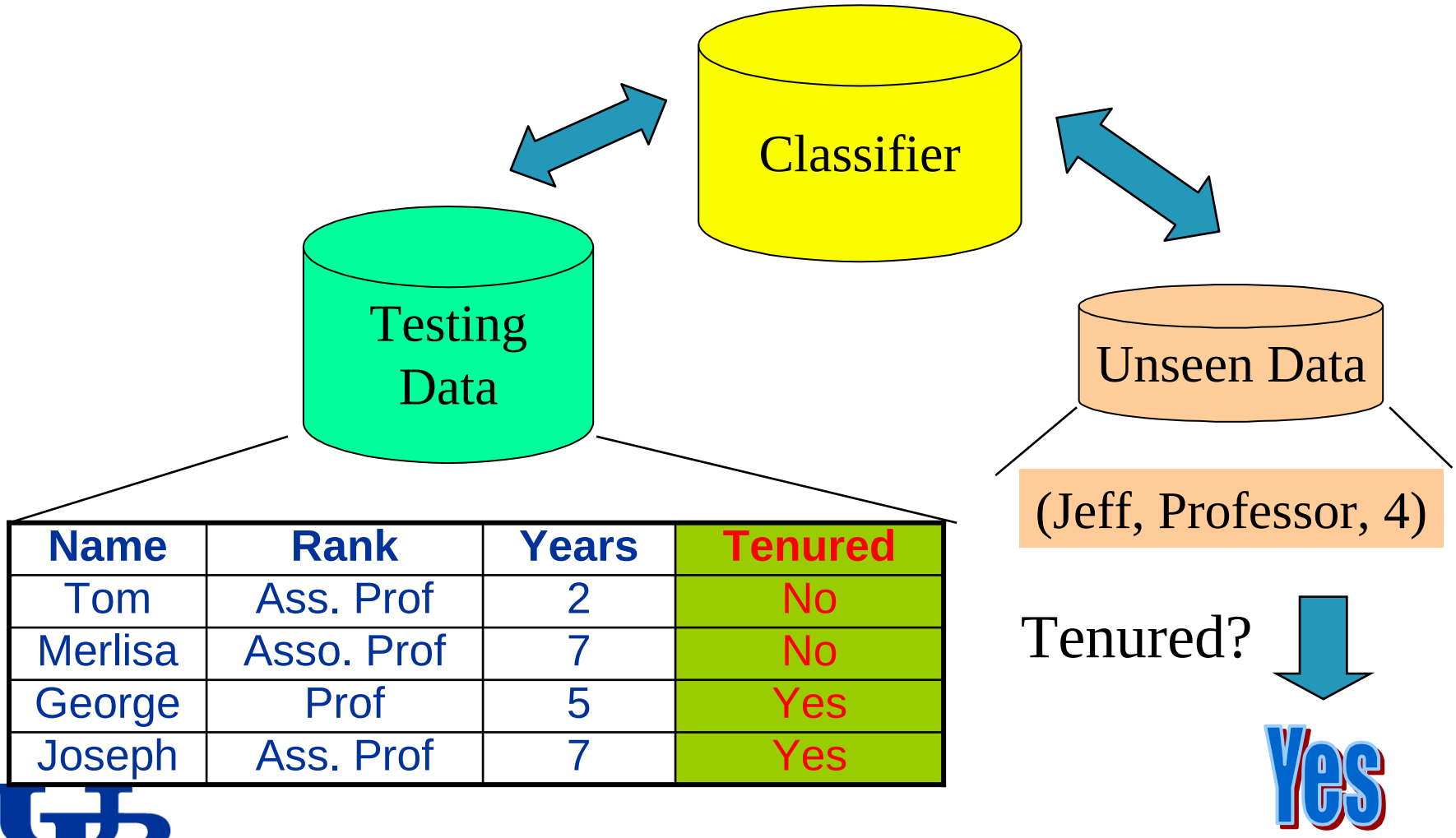
- Model construction: describe a set of predetermined classes
  - Training dataset: tuples for model construction, each tuple/sample belongs to a predefined class
  - Classification rules, decision trees, or math formulae
- Model application: classify unseen objects
  - Estimate accuracy of the model using an independent test set
  - Acceptable accuracy → apply the model to classify data tuples with unknown class labels

Model Construction



Name	Rank	Years	Tenured
Mike	Ass. Prof	3	No
Mary	Ass. Prof	7	Yes
Bill	Prof	2	Yes
Jim	Asso. Prof	7	Yes
Dave	Ass. Prof	6	No
Anne	Asso. Prof	3	No

Model Application



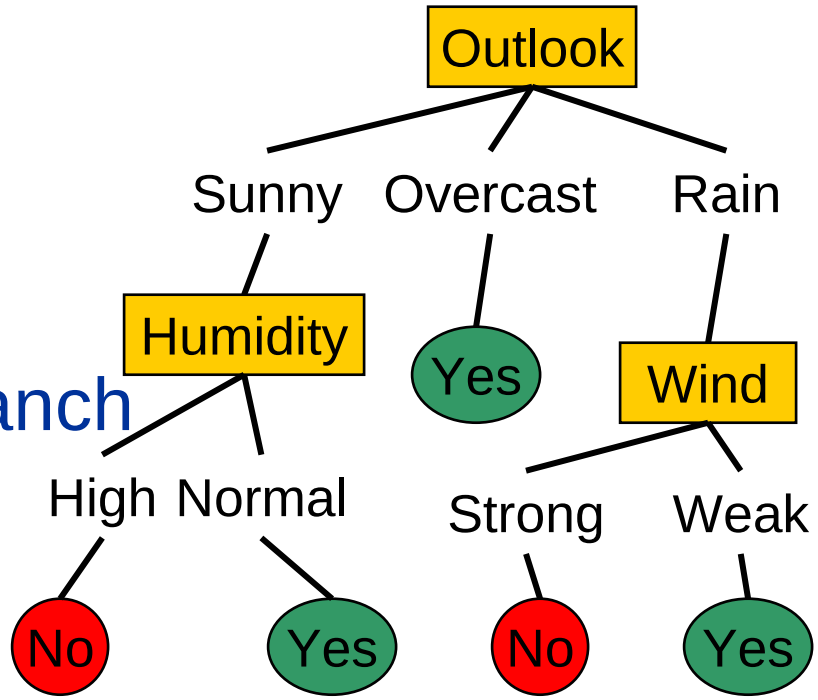
Name	Rank	Years	Tenured
Tom	Ass. Prof	2	No
Merlisa	Asso. Prof	7	No
George	Prof	5	Yes
Joseph	Ass. Prof	7	Yes

Supervised vs. Unsupervised Learning

- Supervised learning (classification)
  - Supervision: objects in the training set have labels
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Decision Tree

- A node in the tree: a test of some attribute
- A branch: a possible value of the attribute
- Classification
  - Start at the root
  - Test the attribute
  - Move down the tree branch



Basic Algorithm ID3

- Construct a tree in a top-down recursive divide-and-conquer manner
  - Which attribute is the best at the current node?
  - Create a nodes for each possible attribute value
  - Partition training data into descendant nodes
- Conditions for stopping recursion
  - All samples at a given node belong to the same class
  - There is no attribute remaining for further partitioning – majority voting is employed for classifying the leaf
  - There is no sample at the node

Which Attribute Is The Best?

- Select the attribute that is most useful for classifying examples
- Information gain and gini index
  - Statistical properties
  - Measure how well an attribute separates the training examples

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Extracting Classification Rules From Decision Trees

- Each path from the root to a leaf → an IF-THEN rule
  - Each attribute-value pair along a path forms a conjunction
  - The leaf node holds the class prediction
  - IF age = “<=30” AND student = “no” THEN buys_computer = “no”
- Rules are easy to understand

Some Other Classification Methods

- Neural networks
- Bayesian classification
  - Naïve Bayesian classification
  - Bayesian belief network
- Support vector machines

Classification for Intrusion Detection

- Misuse detection
  - Classification based on known intrusions
- Example: Sinclair et al. “An application of machine learning to network intrusion detection”
  - Use decision trees and ID3 on host session data
  - Use genetic algorithms to generate rules
    - If <pattern> then <alert>

HIDE

- "A hierarchical network intrusion detection system using statistical processing and neural network classification" by Zheng et al.
- Five major components
  - Probes collect traffic data
  - Event preprocessor preprocesses traffic data and feeds the statistical model
  - Statistical processor maintains a model for normal activities and generates vectors for new events
  - Neural network classifies the vectors of new events
  - Postprocessor generates reports

Modeling Worm Propagation

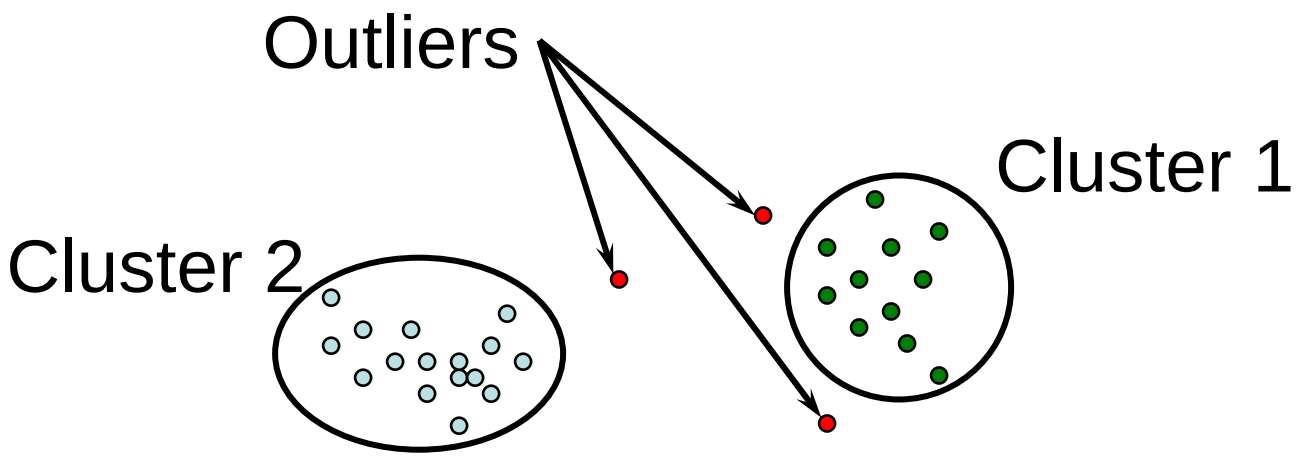
- C.C. Zou et al, CCS'02
- Model the Code Read propagation
  - Based on the classical epidemic Kermack-Mckendrick model
- The two-factor worm model: understand and predict the scale and speed of Internet worm spreading
  - The dynamic countermeasures taken by ISPs and users
  - The slowed down worm infection rate because Code Red rampant propagation caused congestion and troubles to some routers

Intrusion Detection by NN and SVM

- S. Mukkamala et al., IEEE IJCNN May 2002
- Discover useful patterns or features that describe user behavior on a system
- Use the set of relevant features to build classifiers
- SVMs have great potential to be used in place of NNs due to its scalability and faster training and running time
- NNs are especially suited for multi-category classification

What Is Clustering?

- Group data into clusters
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
  - Unsupervised learning: no predefined classes



What Is A Good Clustering?

- High intra-class similarity and low inter-class similarity
  - Depending on the similarity measure
- The ability to discover some or all of the hidden patterns

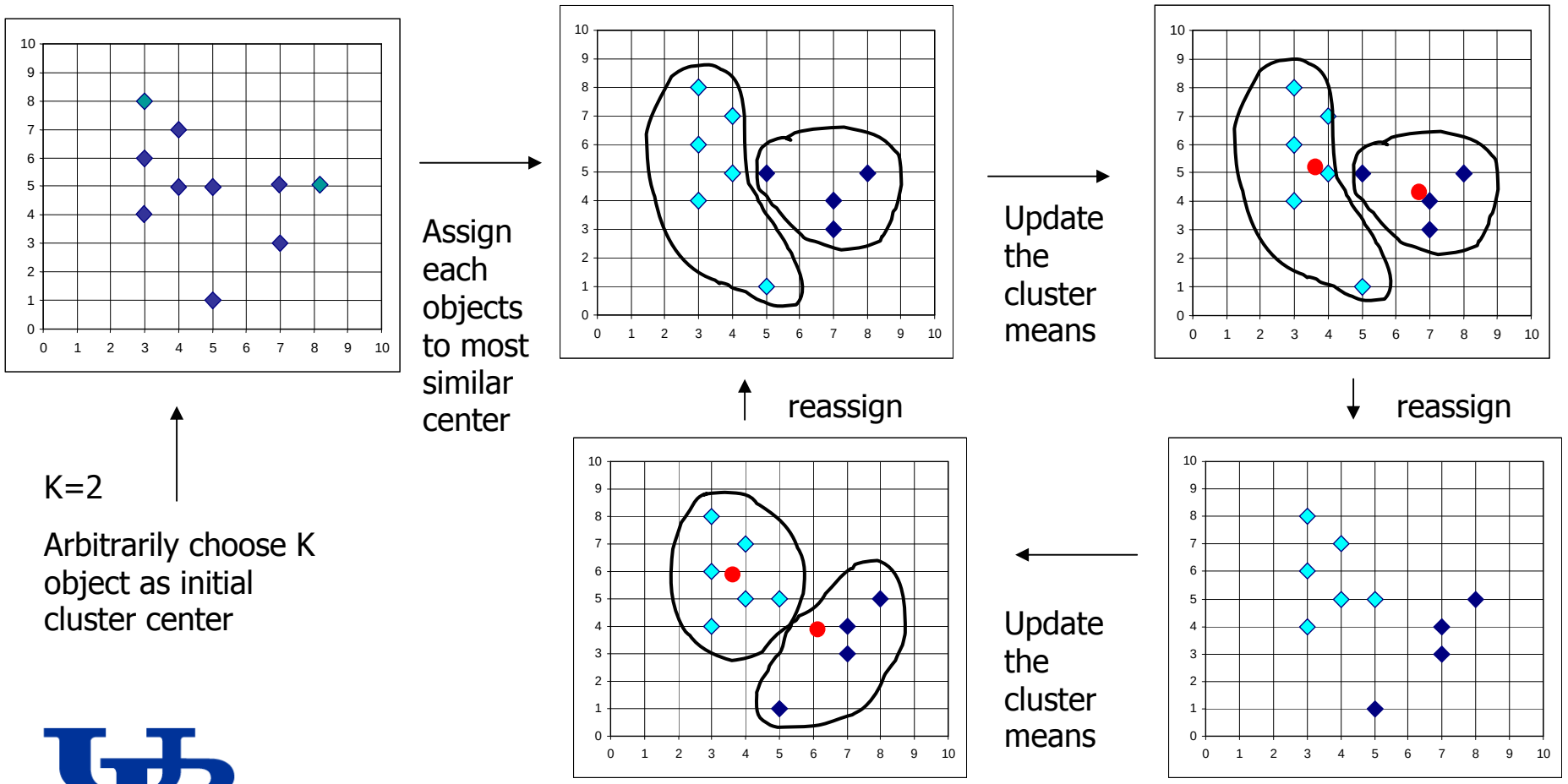
Requirements of Clustering

- Able to handle various types of attributes, high dimensional data, noise and outliers
- Insensitive to order of input records
- Scalable w.r.t. data set size and dimensionality
- Discover clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Good interpretability and usability

Clustering Approaches

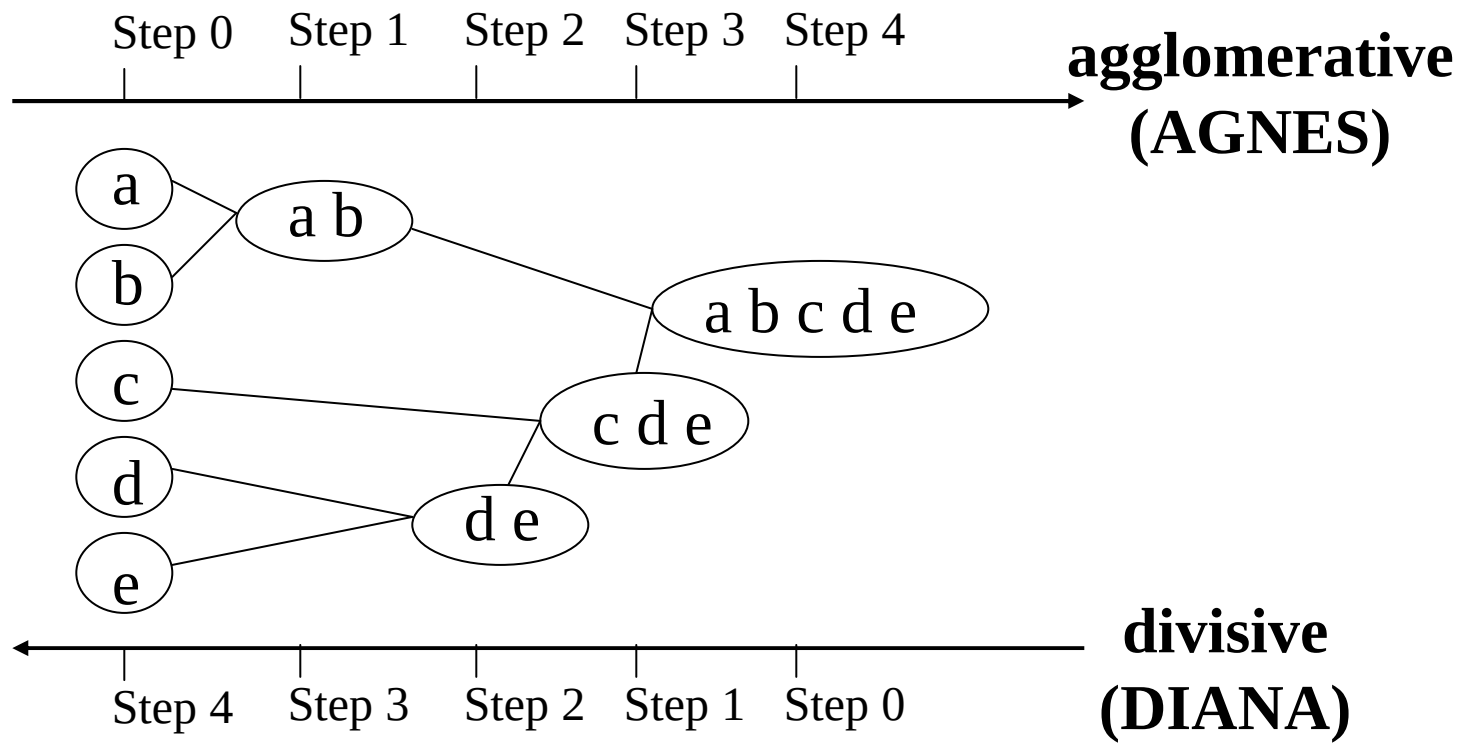
- Partitioning algorithms
  - Partition the objects into k clusters
  - Iteratively reallocate objects to improve the clustering
- Hierarchy algorithms
  - Agglomerative: each object is a cluster, merge clusters to form larger ones
  - Divisive: all objects are in a cluster, split it up into smaller clusters

K-Means: Example



Hierarchical Clustering

- Group data objects into t tree of clusters

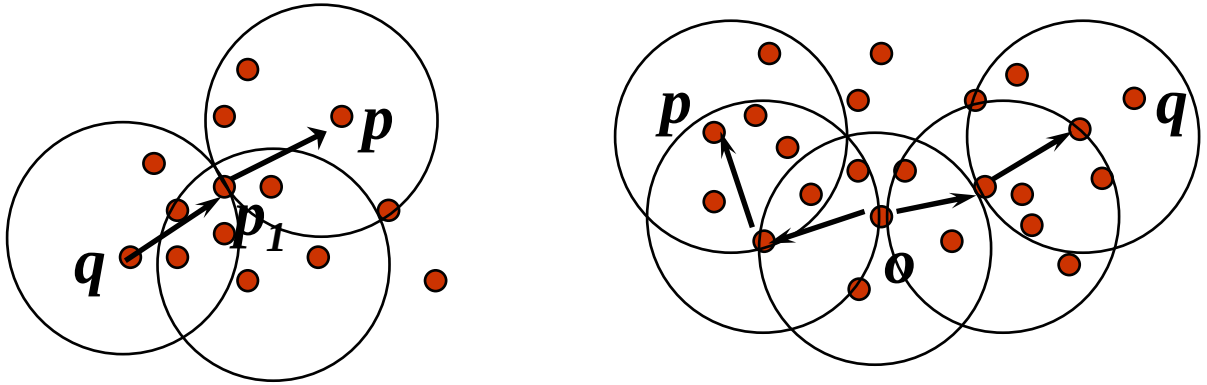


More Clustering Approaches

- Density-based methods
  - Based on connectivity and density functions
  - Filter out noise, find clusters of arbitrary shape
- Grid-based methods
  - Quantize the object space into a grid structure
- Model-based
  - Use a model to find the best fit of data

Density-Reachable / Connected

- Density-reachable
  - Directly density reachable $p_1 \rightarrow p_2$, $p_2 \rightarrow p_3$, …, $p_{n-1} \rightarrow p_n$ ➔ p_n density-reachable from p_1
- Density-connected
  - Points p, q are density-reachable from o ➔ p and q are density-connected



Outlier Analysis

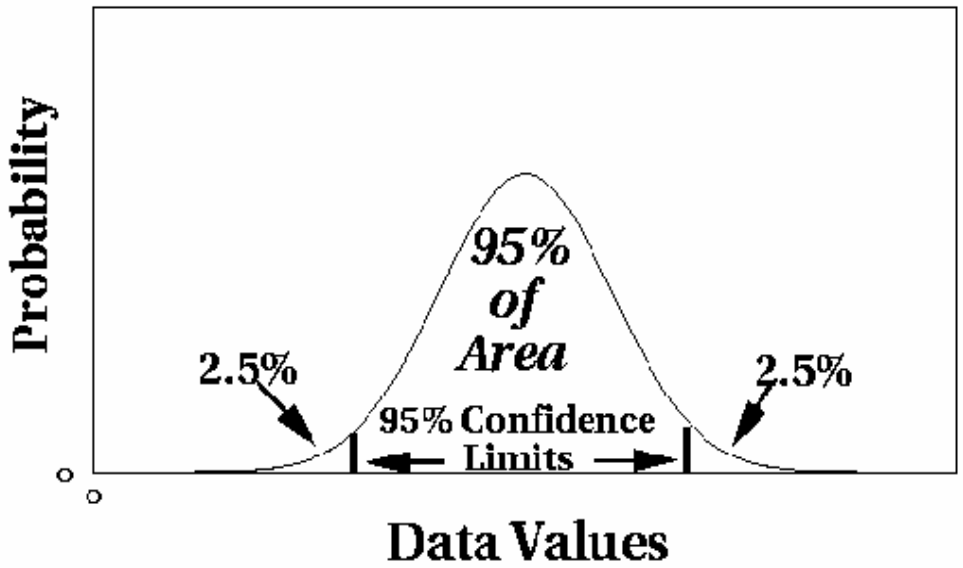
- "*One person's noise is another person's signal*"
- Outliers: the objects considerably dissimilar from the remainder of the data
  - Examples: intrusions, credit card fraud, Michael Jordon, intrusions, etc
  - Applications: intrusion detection, credit card fraud detection, telecom fraud detection, customer segmentation, medical analysis, etc

Statistical Outlier Analysis

- Discordancy/outlier tests
  - 100+ tests proposed
- Data distribution
  - Distribution parameters
- The number of outliers
- The types of expected outliers
  - Example: upper or lower outliers in an ordered sample

Statistical Approaches

- Most tests are univariate
  - Hard for multidimensional datasets
- All are distribution-based
  - Unknown distributions in many applications



Distance-based Outliers

- A DB(p, D)-outlier is an object O in a dataset T s.t. at least fraction p of the objects in T lies at a distance greater than distance D from O
- Algorithms for mining distance-based outliers
  - The index-based algorithm, the nested-loop algorithm, the cell-based algorithm

Clustering for Intrusion Detection

- Anomaly detection
  - Any significant deviations from the expected behavior are reported as possible attacks
- Build clusters as models for normal activities
- "A scalable clustering for intrusion signature recognition" by Ye and Li
  - Use description of clusters as signatures of intrusions

Using Artificial Anomalies

- Fan et al., in IEEE ICDM'01
- The boundary between known classes and anomalies learned from real data is often vague
- Generate artificial anomalies to coerce the inductive learner into discovering an accurate boundary between normal connections and known intrusions and anomalies

Alert Correlation

- F. Cuppens and A. Miege, in IEEE S&P'02
- Use clustering and merging functions to recognize alerts that correspond to the same occurrence of an attack
  - Create a new alert that merge data contained in these various alerts
- Generate global and synthetic alerts to reduce the number of alerts further

Outlier Analysis for Intrusion Detection

- Outliers may correspond to attacks
- PHAD and ALAD
  - PHAD for Ethernet, IP and transport layer packet headers
  - ALAD for TCP data
  - Use fields from headers as features
  - Cluster attack-free data
  - A new packet not in any cluster is an outlier (anomaly), an anomaly score is calculated

Mining Data Streams

- Continuous arrival data in multiple, rapid, time-varying, possibly unpredictable and unbounded streams
- Many applications
  - Financial applications, network monitoring, security, telecommunications data management, web application, manufacturing, sensor networks, etc.

Classification Data Streams

- Concept drifting – the data distribution may change over time
- Can we maintain an accurate model for the data?
  - Incremental maintenance of a data model (e.g., a decision tree) over a data stream
  - The maintenance has to be quick for fast streams and robust for noisy data

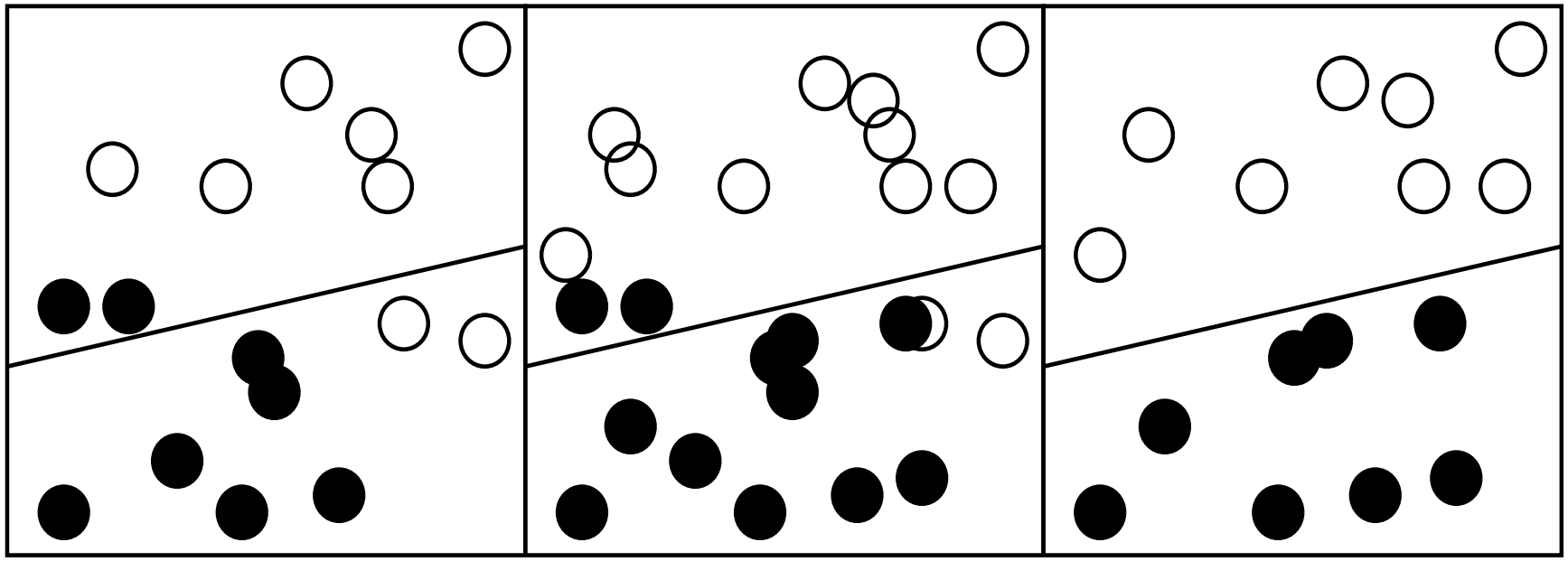
Ensemble Classifiers

- Maintaining an accurate classifier can be costly
- Use the (linear) combination of a subset of classifiers to approximate the model
  - Build a set of decision trees for data at some instants
  - For new data, check if some combination works good enough
  - Build new model only if ensemble of classifiers fails

Ensemble Classifiers

- Wang et al. KDD'03

optimum boundary: ——



(a) S_2+S_1 (b) $S_2+S_1+S_0$ (c) S_2+S_0

Clustering Data Streams

- Maintain quality clustering over a data stream
  - Clusters may change in shape and distribution
  - Quality guarantee
- "You only get one look"

A K-medians Method [Guha et al.]

- For a period, reduce the raw data to *O(k)* intermediate medians with weights
- Cluster intermediate medians to get maintain global clusters
- With quality guarantee

Clustering Streams in Two Phases

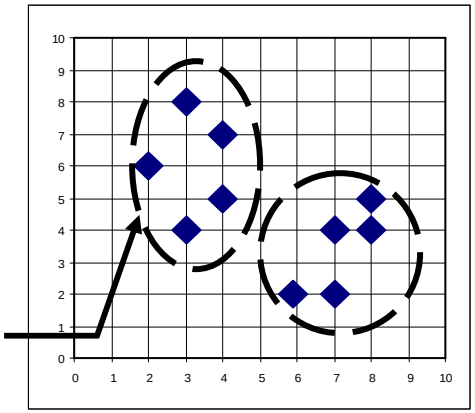
- Phase 1: summarize a data stream into micro-clusters
  - Maintain clustering feature vectors
  - Clustering feature vectors are addible
- Phase 2: clustering analysis based on micro-clusters

Clustering Feature

- Clustering feature $CF = (N, \vec{LS}, SS)$
  - *N*: number of objects
  - $\vec{LS} = \sum_{o} \vec{o}$
  - $SS = \sum_{o} o^2$

(3,4), (2,6), (4,5), (4,7), (3,8)

CF = (5, (16,30),(54,190))



- Clustering features are addible
  - $CF_1 + CF_2 = (N_1 + N_2, \vec{LS}_1 + \vec{LS}_2, SS_1 + SS_2)$

Mining Data Streams for Intrusion Detection

- Maintaining profiles of normal activities
  - The profiles of normal activities may drift
- Identifying novel attacks
  - Identifying clusters and outliers in traffic data streams
- Reduce the future alarm load by writing filtering rules that automatically discard well-understood false positives

Spatial-temporal Mining

- Consider the location and time of the data objects during the mining
- Detecting routing-based attacks – V. Mittal and G. Vigna, CCS'02
  - Malicious routing behavior can be identified only in specific network locations
- Use information about both the network topology and the positions of sensors
  - Automatically generate the appropriate sensor signatures

Integrating Multiple Mining Methods

- W. Lee et al., in Information and System Security
- Use auditing programs to extract an extensive set of features
- Apply data mining methods to learn rules
  - Classification, meta-learning, association rules, and frequent episodes
- Work for misuse detection and anomaly detection

Conclusions

- Intrusion detection is a critical application
- Bad (?) news: grand challenges to achieve effective, accurate and efficient detection
- Good news: data mining can help
- Data mining for intrusion detection is still in its infancy
  - Many interesting and promising research topics

References – Books

- Northcutt and Novak, Network Intrusion Detection (3rd edition), New Riders, 2003.
- Barbará and Jajodia, Applications of Data Mining in Computer Security, Kluwer, 2002.
- Han and Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2001.
- Hand, Mannila and Smyth, Principles of Data Mining, MIT Press, 2001.

References – Recent Tutorials

- Brodley and Chan, Data mining for computer security, ACM KDD'03 tutorial
- Lazarevic, Srivastava and Kuman, Data mining for computer security applications, IEEE ICDM'03 tutorial
- Clifton, Privacy preserving data mining, ACM KDD'03 tutorial
- Pei, Upadhyaya, Farooq and Govindaraju, Data Mining for Intrusion Detection: Techniques, Applications and Systems, ICDE'04 tutorial

References – Some Data Sets

- DARPA 1998 data set
  - A cleansed set in KDDCup'99
  - DARPA 1991 data set is also available
  - http://www.ll.mit.edu/IST/ideval/data/data_index.html
- System call traces data set from University of New Mexico
  - http://www.cs.unm.edu/~immsec/systemcalls.htm
- Solaris audit data using BSM
- MOAT and Auckland II from University of Melbourne, Australia
- Data set with virus files from Columbia University
  - http://www.cs.columbia.edu/ids/mef/software

References – Forums

- ACM International Conference on Computer Security
- IEEE Symposium on Security and Privacy (S&P)
- National Computer Security Conference
- National Information Security Conference
- Data mining conferences
  - ACM KDD, IEEE ICDM, SIAM Data Mining, SIGMOD, VLDB, ICDE, EDBT
- Specific workshops in related conferences

References – Some Recent Papers

- F. Cuppens and A. Miege. Alert correlation in a cooperative intrusion detection framework, in IEEE S&P'02.
- B. Dutertre et al. Intrusion-tolerant enclaves, in IEEE S&P'02.
- W. Fan et al. Using artificial anomalies to detect unknown and known network intrusions, in IEEE ICDM'01.
- C. Kruegel and G. Vigna. Anomaly detection of web-based attacks, in ACM CCS'03.
- W. Lee et al. A data mining framework for building intrusion detection models. In Information and System Security, Vol. 3, No. 4, 2000.
- V. Mittal and G. Vigna. Sensor-based intrusion detection for intra-domain distance-vector routing, in ACM CCS'02.

References – Some Recent Papers (2)

- S. Mukkamala et al. Intrusion detection using neural networks and support vector machines, in IEEE IJCNN (May 2002).
- P. Ning et al. Constructing attack scenarios through correlation of intrusion alerts, in ACM CCS'02.
- R. Sekar et al. Specification-based anomaly detection: a new approach for detecting network intrusions, in ACM CCS'02.
- R. Sommer and V. Paxson. Enhancing byte-level network intrusion detection signatures with context, in ACM CCS'03.
- C.C. Zou et al. Code Red Worm propagation modeling and analysis, in ACM CCS'02.

Thank You!

http://www.cse.buffalo.edu/faculty/jianpei/