



Protected Estimation Of Guideline Classification Under Physical Assault

B.RENUKA BUSHAN

M.Tech Student, Dept.of computer science
Engineering, Sanketika vidya parisad engineering
college, Pm palem, Visakhapatnam

AJ LAVANYA

Assistant professor, Dept.of computer science
Engineering, Sanketika vidya parisad engineering
college, Pm palem, Visakhapatnam

Abstract; Pattern classification systems are commonly used in adversarial applications, like biometric authentication, network intrusion detection, and spam filtering, in which data can be purposely manipulated by humans to undermine their operation. As this adversarial scenario is not taken into account by classical design methods, pattern classification systems may exhibit vulnerabilities, whose exploitation may severely affect their performance, and consequently limit their practical utility. Extending pattern classification theory and design methods to adversarial settings is thus a novel and very relevant research direction, which has not yet been pursued in a systematic way. In this paper, we address one of the main open issues: evaluating at design phase the security of pattern classifiers, namely, the performance degradation under potential attacks they may incur during operation. We propose a framework for empirical evaluation of classifier security that formalizes and generalizes the main ideas proposed in the literature, and give examples of its use in three real applications. Reported results show that security evaluation can provide a more complete understanding of the classifier's behavior in adversarial environments, and lead to better design choices

I. INTRODUCTION

What is Data Mining?

Structure of Data Mining

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

How Data Mining Works?

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks. Generally, any of four types of relationships are sought:

- Classes: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.

- Clusters: Data items are grouped according to logical relationships or consumer preferences. For

example, data can be mined to identify market segments or consumer affinities.

- Associations: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.

- Sequential patterns: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

Data mining consists of five major elements:

- 1)Extract, transform, and load transaction data onto the data warehouse system.

- 2)Store and manage the data in a multidimensional database system.

- 3)Provide data access to business analysts and information technology professionals.

- 4)Analyze the data by application software.

- 5)Present the data in a useful format, such as a graph or table.

Different levels of analysis are available:

- Artificial neural networks: Non-linear predictive models that learn through training and resemble biological neural networks in structure.

- Genetic algorithms: Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.

- Decision trees: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID.

- Nearest neighbor method: A technique that classifies each record in a dataset based on a combination of the classes of the k record(s) most similar to it in a historical dataset (where k=1). Sometimes called the k-nearest neighbor technique.

- Rule induction: The extraction of useful if-then rules from data based on statistical significance.

- Data visualization: The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

Characteristics of Data Mining:

- Large quantities of data: The volume of data so great it has to be analyzed by automated techniques e.g. satellite information, credit card transactions etc.

- Noisy, incomplete data: Imprecise data is the characteristic of all data collection.

- Complex data structure: conventional statistical analysis not possible

- Heterogeneous data stored in legacy systems

Benefits of Data Mining:

1)It's one of the most effective services that are available today. With the help of data mining, one can discover precious information about the customers and their behavior for a specific set of products and evaluate and analyze, store, mine and load data related to them

2)An analytical CRM model and strategic business related decisions can be made with the help of data mining as it helps in providing a complete synopsis of customers

3)An endless number of organizations have installed data mining projects and it has helped them see their own companies make an unprecedented improvement in their marketing strategies (Campaigns)

4)Data mining is generally used by organizations with a solid customer focus. For its flexible nature as far as applicability is concerned is being used

II. LITERATURE SURVEY

1) Robustness of Multimodal Biometric Fusion Methods against Spoof Attacks

AUTHORS: R.N. Rodrigues, L.L. Ling, and V. Govindaraju

In this paper, we address the security of multimodal biometric systems when one of the modes is successfully spoofed. We propose two novel fusion schemes that can increase the security of multimodal biometric systems. The first is an extension of the likelihood ratio based fusion scheme and the other uses fuzzy logic. Besides the matching score and sample quality score, our proposed fusion schemes also take into account the intrinsic security of each biometric system being fused. Experimental results have shown that the proposed methods are more robust against spoof attacks when compared with traditional fusion methods

2) Multimodal Fusion Vulnerability to Non-Zero Effort (Spoof) Imposters

AUTHORS: P. Johnson, B. Tan, and S. Schuckers

In biometric systems, the threat of “spoofing”, where an imposter will fake a biometric trait, has led to the increased use of multimodal biometric systems. It is assumed that an imposter must spoof all modalities in the system to be accepted. This paper looks at the cases where some but not all modalities are spoofed. The contribution of this paper is to outline a method for assessment of multimodal systems and underlying fusion algorithms. The framework for this method is described and experiments are conducted on a multimodal database of face, iris, and fingerprint match scores.

III.IMPLEMENTATION

MODULES:

1. Attack Scenario and Model of the Adversary
2. Pattern Classification
3. Adversarial classification:
4. Security modules

MODULES DESCRIPTION:

Attack Scenario and Model of the Adversary:

Although the definition of attack scenarios is ultimately an application-specific issue, it is possible to give general guidelines that can help the designer of a pattern recognition system. Here we

propose to specify the attack scenario in terms of a conceptual model of the adversary that encompasses, unifies, and extends different ideas from previous work. Our model is based on the assumption that the adversary acts rationally to attain a given goal, according to her knowledge of the classifier, and her capability of manipulating data. This allows one to derive the corresponding optimal attack strategy.

Pattern Classification:

Multimodal biometric systems for personal identity recognition have received great interest in the past few years. It has been shown that combining information coming from different biometric traits can overcome the limits and the weaknesses inherent in every individual biometric, resulting in a higher accuracy. Moreover, it is commonly believed that multimodal systems also improve security against Spoofing attacks, which consist of claiming a false identity and submitting at least one fake biometric trait to the system (e.g., a “gummy” fingerprint or a photograph of a user’s face). The reason is that, to evade multimodal system, one expects that the adversary should spoof all the corresponding biometric traits. In this application example, we show how the designer of a multimodal system can verify if this hypothesis holds, before deploying the system, by simulating spoofing attacks against each of the matchers.

Adversarial classification:

Assume that a classifier has to discriminate between legitimate and spam emails on the basis of their textual content, and that the bag-of-words feature representation has been chosen, with binary features denoting the occurrence of a given set of words

Security modules:

Intrusion detection systems analyze network traffic to prevent and detect malicious activities like intrusion attempts, ROC curves of the considered multimodal biometric system under a simulated spoof attack against the fingerprint or the face matcher. Port scans, and denial-of-service attacks. When suspected malicious traffic is detected, an alarm is raised by the IDS and subsequently handled by the system administrator. Two main kinds of IDSs exist: misuse detectors and anomaly-based ones. Misuse detectors match the analyzed network traffic against a database of signatures of known malicious activities. The main drawback is that they are not able to detect never-before-seen malicious activities, or even variants of known ones. To overcome this issue, anomaly-based detectors have been proposed. They build a statistical model of the normal traffic using machine learning techniques, usually one-class

classifiers, and raise an alarm when anomalous traffic is detected. Their training set is constructed, and periodically updated to follow the changes of normal traffic, by collecting unsupervised network traffic during operation, assuming that it is normal (it can be filtered by a misuse detector, and should)

Software Environment

Java Technology

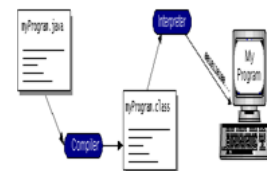
Java technology is both a programming language and a platform.

The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

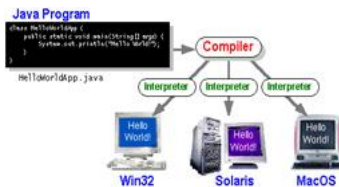
- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java byte codes* —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it’s a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere”

possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



IV. SYSTEM ANALYSIS

EXISTING SYSTEM:

Pattern classification systems based on classical theory and design methods do not take into account adversarial settings; they exhibit vulnerabilities to several potential attacks, allowing adversaries to undermine their effectiveness. A systematic and unified treatment of this issue is thus needed to allow the trusted adoption of pattern classifiers in adversarial environments, starting from the theoretical foundations up to novel design methods, extending the classical design cycle of. In particular, three main open issues can be identified: (i) analyze the vulnerabilities of classification algorithms, and the corresponding attacks. (ii) Developing novel methods to assess classifier security against these attacks, which are not possible using classical performance evaluation methods. (iii) Developing novel design methods to guarantee classifier security in adversarial environments.

DISADVANTAGES OF EXISTING SYSTEM:

1. Poor analyzing the vulnerabilities of classification algorithms, and the corresponding attacks.
2. A malicious webmaster may manipulate search engine rankings to artificially promote website.

V. PROPOSED SYSTEM:

In this work we address issues above by developing a framework for the empirical evaluation of classifier security at design phase that extends the model selection and performance evaluation steps of the classical design cycle. We summarize previous work, and point out three main ideas that emerge from it. We then formalize and generalize them in our framework. First, to pursue security in the context of an arms race it is not sufficient to react to observed attacks, but it is also necessary to proactively anticipate the adversary by predicting the most relevant, potential attacks through a what-if analysis; this allows one to develop suitable

countermeasures before the attack actually occurs, according to the principle of security by design. Second, to provide practical guidelines for simulating realistic attack scenarios, we define a general model of the adversary, in terms of her goal, knowledge, and capability, which encompass and generalize models proposed in previous work. Third, since the presence of carefully targeted attacks may affect the distribution of training and testing data separately, we propose a model of the data distribution that can formally characterize this behaviour, and that allows us to take into account a large number of potential attacks; we also propose an algorithm for the generation of training and testing sets to be used for security evaluation, which can naturally accommodate application-specific and heuristic techniques for simulating attacks.

ADVANTAGES OF PROPOSED SYSTEM:

1. Proposed system prevents developing novel methods to assess classifier security against these attacks.
2. The presence of an intelligent and adaptive adversary makes the classification problem highly non-stationary.

VI. SYSTEM DESIGN

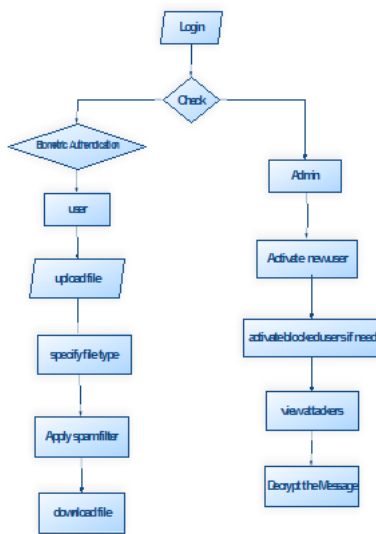
SYSTEM ARCHITECTURE:



DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at

any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



VII. UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize

of instant. Thus the objective of input design is to create an input layout that is easy to follow

VIII. SCREEN SHOTS





OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

IX. CONCLUSION

In this paper we focused on empirical security evaluation of pattern classifiers that have to be deployed in adversarial environments, and proposed how to revise the classical performance evaluation design step, which is not suitable for this purpose.

Our main contribution is a framework for empirical security evaluation that formalizes and generalizes ideas from previous work, and can be applied to different classifiers, learning algorithms, and classification tasks. It is grounded on a formal model of the adversary, and on a model of data distribution that can represent all the attacks considered in previous work; provides a systematic method for the generation of training and testing sets that enables security evaluation; and can accommodate application-specific techniques for attack simulation. This is a clear advancement with respect to previous work, since without a general framework most of the proposed techniques (often tailored to a given classifier model, attack, and application) could not be directly applied to other problems.

An intrinsic limitation of our work is that security evaluation is carried out empirically, and it is thus datadependent; on the other hand, model-driven analyses [12], [17], [38] require a full analytical model of the problem and of the adversary's behavior, that may be very difficult to develop for real-world applications. Another intrinsic limitation is due to fact that our method is not application-specific, and, therefore, provides only high-level guidelines for simulating attacks. Indeed, detailed guidelines require one to take into account application-specific constraints and adversary models. Our future work will be devoted to develop techniques for simulating attacks for different applications.

Although the design of secure classifiers is a distinct problem than security evaluation, our framework could be also exploited to this end. For instance, simulated attack samples can be included into the training data to improve security of

discriminative classifiers (e.g., SVMs), while the proposed data model can be exploited to design more secure generative classifiers. We obtained encouraging preliminary results on this topic

X. REFERENCES

- [1] R.N. Rodrigues, L.L. Ling, and V. Govindaraju, "Robustness of Multimodal Biometric Fusion Methods against Spoof Attacks," *J. Visual Languages and Computing*, vol. 20, no. 3, pp. 169-179, 2009.
- [2] P. Johnson, B. Tan, and S. Schuckers, "Multimodal Fusion Vulnerability to Non-Zero Effort (Spoof) Imposters," *Proc. IEEE Int'l Workshop Information Forensics and Security*, pp. 1-5, 2010.
- [3] P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee, "Polymorphic Blending Attacks," *Proc. 15th Conf. USENIX Security Symp.*, 2006.
- [4] G.L. Wittel and S.F. Wu, "On Attacking Statistical Spam Filters," *Proc. First Conf. Email and Anti-Spam*, 2004.
- [5] D. Lowd and C. Meek, "Good Word Attacks on Statistical Spam Filters," *Proc. Second Conf. Email and Anti-Spam*, 2005.
- [6] A. Kolcz and C.H. Teo, "Feature Weighting for Improved Classifier Robustness," *Proc. Sixth Conf. Email and Anti-Spam*, 2009.
- [7] D.B. Skillicorn, "Adversarial Knowledge Discovery," *IEEE Intelligent Systems*, vol. 24, no. 6, Nov./Dec. 2009.
- [8] D. Fetterly, "Adversarial Information Retrieval: The Manipulation of Web Content," *ACM Computing Rev.*, 2007.
- [9] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*. Wiley-Interscience Publication, 2000.
- [10] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial Classification," *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 99-108, 2004.
- [11] M. Barreno, B. Nelson, R. Sears, A.D. Joseph, and J.D. Tygar, "Can Machine Learning be Secure?" *Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS)*, pp. 16-25, 2006.
- [12] A.A. C_ardenas and J.S. Baras, "Evaluation of Classifiers: Practical Considerations for Security Applications," *Proc. AAAI Workshop Evaluation Methods for Machine Learning*, 2006.

- [13] P. Laskov and R. Lippmann, "Machine Learning in Adversarial Environments," *Machine Learning*, vol. 81, pp. 115-119, 2010.

AUTHOR's PROFILE



B.RENUKA BUSHAN, M.Tech Student, Dept.of computer science Engineering, Sanketika vidya parisad engineering college, Pmpalem, Visakhapatnam



AJ LAVANYA, Assistant professor, Dept.of computer science Engineering, Sanketika vidya parisad engineering college, Pmpalem, Visakhapatnam.