LA-13075-MS

Knowledge Fusion: Time Series Modeling Followed by Pattern Recognition Applied to Unusual Sections of Background Data

Los Alamos
NATIONAL LABORATORY

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

Edited by Paul W. Henriksen, Group CIC-1
Prepared by Sharon Hurdle, Group NIS-7

This work was supported by the U.S. Department of Energy, Office of Nonproliferation and National Security.

An Affirmative Action/Equal Opportunity Employer

LA-13075-MS

UC-700
Issued: March 1996

Knowledge Fusion: Time Series Modeling Followed by Pattern Recognition Applied to Unusual Sections of Background Data

Tom Burr
Justin Doak
Jo Ann Howell
Dave Martinez
Richard Strittmatter

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

Executive Summary

This report describes work performed during FY 95 for the Knowledge Fusion Project, which was sponsored by the Department of Energy, Office of Nonproliferation and National Security. The project team selected satellite sensor data as the one main example to which its analysis algorithms would be applied. The specific sensor-fusion problem has many generic features that make it a worthwhile problem to attempt to solve in a general way. The generic problem is to recognize events of interest from multiple time series in a possibly noisy background. By implementing a suite of time series modeling and forecasting methods and using well-chosen alarm criteria, we reduce the number of false alarms. We then further reduce the number of false alarms by analyzing all suspicious sections of data, as judged by the alarm criteria, with pattern recognition methods. This report describes the implementation and application of this two-step process for separating events from unusual background. As a fortunate by-product of this activity, it is possible to gain a better understanding of the natural background.

The fundamental principle underlying the DOE/NN-sponsored Knowledge Fusion (KF) project at Los Alamos is that the technologies of extracting information from huge data sets can best be developed by cooperation between a team of specialists experienced with the general underlying computer science and mathematical problems and experts in the design and operation of specific operational systems. The goal is to solve the specific problems so that the results will be as generally applicable as possible to new problem domains as they develop. This goal includes the development of a computational environment that will insure that all tools developed for tasks such as parsing, filtering, analyzing, and displaying can be used within a unified environment that encourages easy application to new situations. Results presented here will be primarily from a specific example problem in which several sensors on satellites are used to detect nuclear weapon explosions. The plan is to solve that particular problem and to solve it in a computational environment that will permit ready application of the tools developed to new domains. If successful, many DOE projects will eventually benefit, and a uniform method of collection, storage, and analysis should emerge as the benefits of using the knowledge fusion team become apparent.

More specifically, this report presents a somewhat detailed application of 12 different time-series forecasting methods, followed by 7 different pattern recognition methods. One of the 12 forecasting methods and 1 of the 7 pattern recognition methods were written for the KF project. The other forecasting methods and pattern recognition methods were obtained from commercial software implementations, as described in Refs. ES-1 and ES-2. It is not possible to give a complete treatment of all of these candidate methods in one document, so this report has two companion reports. Reference ES-3 gives a detailed comparison of two of the forecasting methods (fuzzy-forecaster and statistically motivated curve smoothers as forecasters), and Ref. ES-4 gives a detailed discussion of the linear forecasting methods, including vector-valued autoregressive moving average (ARMA) time series models. Reference ES-4 also gives considerable detail about one pattern recognition method that was empirically derived from one specific data set. The empirical derivation is closely related to quadratic discriminant analysis.

References

ES.1. S. Bleasdale, T. Burr, A. Coulter, J. Doak, B. Hoffbauer, D. Martinez, J. Prommel, C. Scovel, R. Strittmatter, T. Thomas, and A. Zardecki, "Knowledge Fusion: Analysis of Vector-Based Time Series with an Example from the SABRS Project," Los Alamos National Laboratory report LA-12931-MS (April 1995).

ES-2. T. Burr, A. Coulter, J. Doak, B. Hoffbauer, D. Martinez, and J. Prommel, "Demonstration of the Software Toolkit for Analysis Research," Los Alamos National Laboratory report LA-12924-MS (March 1995).

ES-3. T. Burr and R. Strittmatter, "Knowledge Fusion: Comparison of Fuzzy Curve Smoother to Statistically Motivated Curve Smoothers," Los Alamos National Laboratory report LA-13076-MS (February 1996).

ES-4. S. Bleasdale, T. Burr, C. Scovel, and R. Strittmatter, "Knowledge Fusion: An Approach to Time Series Model Selection Followed by Pattern Recognition," Los Alamos National Laboratory report LA-13095-MS (February 1996).

Contents

Knowledge Fusion: Time Series Modeling Followed by Pattern Recognition Applied to Unusual Sections of Background Data

by

Tom Burr, Justin Doak, Jo Ann Howell, Dave Martinez, and Richard Strittmatter

ABSTRACT

This report describes work during FY 95 that was sponsored by the Department of Energy, Office of Nonproliferation and National Security, for the Knowledge Fusion Project. The project team selected satellite sensor data as the one main example to which its analysis algorithms would be applied. The specific sensor-fusion problem has many generic features, which make it a worthwhile problem to attempt to solve in a general way. The generic problem is to recognize events of interest from multiple time series that define a possibly noisy background. By implementing a suite of time series modeling and forecasting methods and using well-chosen alarm criteria, we reduce the number of false alarms. We then further reduce the number of false alarms by analyzing all suspicious sections of data, as judged by the alarm criteria, with pattern recognition methods. This report describes the implementation and application of this two-step process for separating events from unusual background.

---

1. Introduction

The fundamental principle underlying the Knowledge Fusion project at Los Alamos, sponsored by the Department of Energy, Office of Nonproliferation and National Security, is that the technologies of extracting information from huge data sets can best be developed by cooperation between a team of specialists experienced with the general underlying computer science and mathematical problems *and* experts in the design and operation of specific operational systems. The goal is to solve the specific problems so that the results will be as generally applicable as possible to new problem domains as they develop. This goal includes the development of a computational environment that will insure that all tools developed for tasks such as parsing, filtering,

analyzing, and displaying can be used within a unified environment that encourages easy application to new situations (Ref. 1). Results presented here will be primarily from a specific example in which several sensors on satellites are used to detect nuclear weapon explosions. The plan is to solve that particular problem, and to solve it in a computational environment that will permit ready application of the tools developed to new domains. If successful, many DOE projects will eventually benefit, and a uniform method of collection, storage, and analysis should emerge as the benefits of utilizing the knowledge fusion team become apparent.

This report is organized as follows: Section 2 describes the multiple-sensor problem from which we extrapolate general procedures for data analysis in a particular but common setting, reviews previous related work on the sensor-fusion problem, and describes our computational environment. Section 3 covers some specific data-acquisition issues. Section 4 describes analysis methods for vector-valued time series. Section 5 provides more details about one of the two general types of specific data sets. These additional details relate to our approach to applying pattern recognition methods. Also related to our approach to pattern recognition is the choice of forecasting methods, so in section 5 we compare the results of 12 forecasting methods. Section 6 gives details about the raw and derived data presented to pattern-recognition algorithms and results of cluster analysis on some of the data. Section 7 gives pattern-recognition results for seven methods on one simulated and two real data sets. Section 8 is a summary.

2. The Nudet-Detection Problem

2.1 Background

We emphasize that our intention is not to directly support any project but to solve selected problems from particular projects that are considered top candidates to appear in many DOE projects. This report focuses on the following very typical scenario. The most ambitious goal is to use time series data from multiple sensors to deduce the "state of affairs" during any specified time period. A more reasonable goal is to try to deduce whether a specified time period contains an event of interest. In the nudet-detection problem, nudet, a nuclear blast, is the event of interest. In the case of atmospheric and exo-atmospheric surveillance, we know there can be events such as solar flares that are not the event of interest, but they are also not ordinary background. We will therefore divide the background data into two types: usual and unusual. We further divide "unusual" background into two types: known and fairly well characterized, such as solar flares, and unknown. We can now state **goal one** as:

Use time series data from multiple sensors to label each section of time as

1) ordinary background,
2) unusual background of known type or of unknown type, or
3) section containing the event (nudet) of interest.

Figure 1 shows an overview of our knowledge fusion approach to the nudet-detection application. It identifies the steps in the analysis that are described in the following sections in more detail.

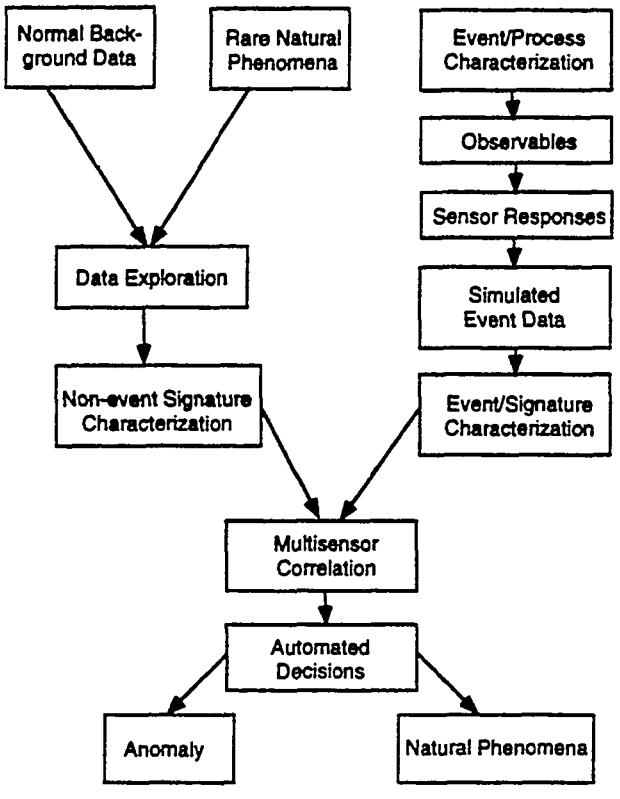

FIGURE 1. Knowledge fusion components as they apply to our nudet-detection problem.

To accomplish goal one, we model the ordinary background to set decision thresholds based on the analysis of data from several sensors. Sections of data that "look unusual" based on decision thresholds are further analyzed by pattern recognition to separate false alarms from true events.

Information on the specific nudet-detection programs can be found in Ref. 2. To keep this report self-contained, we also include here the relevant information.

The United States currently uses particle and radiation detectors aboard satellites to monitor the near-earth environment for clandestine nudets. These satellite-based detectors have access to a telemetry downlink with large bandwidth, and much of the detector output data is transmitted to a ground station for recording and analysis. The on-board electronic systems that control the particle and radiation detectors contain logic elements that can issue alerts and trigger optional recording and data transmission activities when unusual detector signals occur. However, most of the data analysis and anomaly resolution activities occur only after the detector outputs are received by the ground station. Therefore, a potential application of our project is to apply knowledge fusion methods to develop the computational algorithms required for this on-board processing of detector outputs.

Because most of the detectors of interest generate output values at regular intervals, much of the analysis of the satellite detector systems can be formulated as a time-series problem involving multiple time series that have nonzero cross correlations at both zero and nonzero time lags. Consequently, we are evaluating a variety of both standard and innovative analysis methods for time series to determine their effectiveness in correctly interpreting the detector signals. To evaluate these algorithms, we must test them on a large quantity of detector output data corresponding both to natural background processes alone and to natural background processes with a variety of nudet effects superimposed. Some of that work was completed in 1994 and has been applied during 1995. During 1995 we began to add the ability to recognize patterns to all sections of time series that appear unusual based on some alarm criteria.

About four and one-half years of detector output data from the current generation of satellites is available on optical disks at Los Alamos National Laboratory. This is an extensive source of detector response data for natural background processes, covering more than one-third of a sun spot intensity cycle and including a solar activity maximum. We are using these detector output databases to generate data sets for developing and testing our time-series algorithms.

Observational data for actual nudet signals superimposed on natural background processes are much more limited. Several exo-atmospheric nuclear tests were conducted in the HARDTACK, FISHBOWL, and ARGUS programs in the late 1950s and early 1960s, and detector output records from these events are available. However, these data cover a fairly small part of the device-type/detector-location parameter space of interest. In addition, detector designs have changed somewhat since these early tests were conducted. Accordingly, we are using simulation models of nudets and of detectors to generate detector responses to a variety of nudet/detector combinations, and we then add these responses to the background signals from the observational databases to generate the nudet data sets required to develop and evaluate the time-series methods. During 1994 we compared our simulated nudet detector signals with the observational results from the exo-atmospheric tests to assure that the simulation models generate realistic results for the gamma and neutron signals.

The work described in Ref. 2 restricts attention to four time series: electrons, protons, neutrons, and gammas. In Ref. 2, we described two new methods to combine information from the gamma and neutron channels. Forecasting the neutron time series was simple. Forecasting the gamma series was difficult, and open questions remain about the best way to combine information from the gamma and neutron sensors. We also worked toward improving the use of charged particles (electrons and protons) to predict the gammas, but we found that using prior gammas to predict present gammas worked "better." Such a statement can only be partly true, and it depends on the time scales used. See Ref. 2 for more detail, but one key fact is that in Ref. 2 we treated the forecast errors from the neutron series as being independent from the forecast errors from the gamma series during "ordinary background." This is based on physical reasoning and was confirmed with all of the many sections (about ten 1- to 4-hr sections) of ordinary background that we analyzed.

For the work described in this report, we limited attention to three times series—electrons, protons, and gammas—because using the electrons, protons, and prior gammas to forecast present gammas is more challenging than forecasting the neutrons, and because as stated in the previous paragraph, the neutron time series is independent from the gamma time series over periods of ordinary background. Also, during 1995 we were fortunate to gain access to the higher-resolution data sets for the electrons, protons, and gammas that are only available when a threshold criterion has been met. These higher-resolution (more frequent) time series are informally called event records. The current thinking is that each event record must be explained by a human. We randomly selected 1322 event records to apply pattern recognition to separate them into our three categories: nudet, ordinary background, and unusual background. We give more detail in section 6.

To summarize this section, any nudet-detection project requires the use of at least three distinct and widely applicable analysis methodologies:

1. simulation, including both discrete-event simulation and modeling of physical systems;
2. time series analysis; and
3. pattern recognition methods.

Subsequent sections in this report present more detail about the application of these methodologies to any nudet-detection program. It is not difficult to envision many more applications of this general approach. For example, consider nuclear facility monitoring with a central database of count rates from many sensors. Our two-step procedure is still attractive: (1) apply standard or (if needed) innovative methods for modeling and forecasting time series so that the background can be monitored effectively and (2) apply pattern recognition methods to "unusual sections of background data" to explain the source of the unusual behavior.

2.2 The Computational System

More detail is given in Refs. 1 and 2. We continue to develop the software toolkit called Software Toolkit for Analysis Research (STAR). During 1995, a module to perform pattern recognition was added. The pattern-recognition module will be described in the results sections. The goal of the STAR team is to produce a research tool that facilitates the development and interchange of algorithms for locating nonproliferation phenomena of interest in large quantities of data. Using this toolkit, nonproliferation researchers will be able to ascertain which existing techniques are the most promising, develop new and possibly more effective methods, and add/delete algorithms without major re-design work. Some modules or components of STAR will preprocess incoming data; some will select the appropriate information for a particular nonproliferation application; some will analyze data to uncover items of significance to nonproliferation experts; and others will assess the effectiveness of the various components. Some of the specific techniques employed by the various modules will be feature selection algorithms, machine learning algorithms, a pure statistical model, and expert system methodologies.

3. Data Issues

3.1 Real Data

Satellite Data Acquisition. A separate document entitled "GEXO Data Acquisition Guide" (Ref. 3) details the exact steps required to obtain one of the two types of the LANL-based satellite data that we have used. As such, it is both directly relevant to the current application of knowledge fusion and also serves as an example of a typical, real-data problem. Of course, it most importantly serves as a practical working document to transfer the ability to retrieve satellite data to new researchers as required. We refer to the second data type, informally, as the event records. The procedure to access the event records has changed several times, so we have maintained a "how to get the data" report that is unpublished but available from D. Martinez or T. Burr.

3.2 Real Data with Nudets

Fortunately, we do not have examples of real data over the last four years of nudets above the earth's atmosphere. Therefore, we continue to use the simulation code developed during 1994 to create artificial nudet effects that we add to sections of real data. The simulation code includes comprehensive nudet and detector-response modeling, is written in C++, and is accessible in a user-friendly (menu-based) way from within the STAR framework.

4. Analysis of Vector Time Series

In this section we review modeling of vector time series. See Ref. 2 for additional specific details and Refs. 4-6 for general background. There are many published strategies for monitoring changes in the behavior of time series. Nearly all strategies specify that a particular model is in effect for the time series, and the goal is to design tests for departure from that model. Departure is a way to quantify what is meant by a change in behavior of the time series. The new idea here is that all sections of data that depart from the model for ordinary background are included in a data file to be analyzed using clustering and classifying (pattern recognition) methods.

4.1 ARIMA Time Series Models

One widely studied class of time series model is the auto-regressive, integrated moving average (ARIMA) model, which we will use to illustrate the idea. Suppose that the time series $\{X_1, X_2, \ldots, X_n\}$ has been detrended by differencing or, equivalently, by fitting polynomial functions of time. The detrended series is said to follow an auto-regressive, moving average (ARMA) model.

We can write the general scalar *ARMA* (p, q) model as

$$X_t = a_0 + \sum_{j=1}^{p} a_j X_{t-j} + \sum_{j=0}^{q} b_j \varepsilon_{t-j} \tag{1}$$

where the a_j and b_j are real constants, the ε_{t-j} are independently and identically distributed (iid) random variables, and $t \in \{1, 2, \ldots, n\}$. It is common to refer to ε_{t-j} as the shock at time $t-j$. Usually ARIMA models further specify that the shock ε_t follows a Gaussian distribution with mean 0 and variance σ^2, denoted $N(0, \sigma)$. In addition, to ensure that the time series is stationary (constant mean, variance, and covariances) and invertible [representable as an auto-regressive (AR) model], conditions are imposed on the values of the a_j and b_j. See Ref. 6 for further details.

Note that X_t is a linear combination of the past p values of the series (auto-regressive) and of the past q shocks (moving average). We have not restricted this study to the class of linear ARMA models, but because ARMA models are a convenient and easy-to-discuss class of models, most of our discussion will use the ARMA model for an example. Reference 2 introduced 12 candidate ways to model and forecast a vector-based time series. However, at the time Ref. 2 was written, we had no algorithms for vector ARMA time series. All the commercial software had a gap in that they only treated vector AR time series. It is more computationally challenging to handle vector ARMA models. To date, we have heard, but not confirmed, that one lesser-known statistical programming package (JMP) can handle vector ARMA time series. We have added the ability within the statistical and graphical programming language S+ by using a user-contributed library of state-space (in the sense of the Kalman Filter language) functions. Details can be found in Ref. 7.

4.2 Vector AR Models

In this section we give more details about each of the 12 candidate ways to model an AR time series. An AR time series can be written as follows:

$$X_t = f(X_{t-1}, X_{t-2}, \ldots, X_{t-l}) + e_t \,. \tag{2}$$

The error e_t is usually assumed to be from some convenient distribution such as the Gaussian but need not be. Nearly always, the distribution of the e_t is at least assumed to be the same for all t. We will assume that the errors have the same distribution F, which we write as $e_t \sim F(\ )$. Not all functions f combined with error distribution F lead to a stationary time series. We do not attempt a formal treatment of this issue, but rather accept Equation (2) as a basis for trying certain modeling approaches. We also easily generalize (2) to include other time series, say y and Z:

$$X_t = f(X_{t-1}, X_{t-2}, \ldots, X_{t-l_{xxx}}, Y_t, \ldots, Y_{t-l_{yyx}}, Z_t, \ldots, Z_{t-l_z}) + e_t \,. \tag{3}$$

Note that present values of both Y and Z are allowed because the goal is to forecast Z. In our setting, X is the gamma series, Y is the electron series, and Z is the proton series. Or, in another setting, we have only two series: gammas and a series proportional to the sum of electrons and protons.

Equation (3) can be treated exactly like an ordinary regression model for which there are many techniques, loosely described by the degree of assumptions placed on the functional form for f. We will discuss the 12 models in order from the most restrictive assumptions to the least restrictive assumptions made about f. First, we point out two advantages to using restrictive models such as models that assume f is linear.

(1) Often there is not enough data to estimate complicated models or completely unrestricted f. In fact, the old standby, linear regression, will be with us forever for this reason and represents an extreme example of "combining information" in that data at one end of the range is assumed to have the same functional form as data at another end of the range. By the way, because we insist on working with stationary series, we often must restrict the time window so that the series can be considered stationary over that window, thereby reducing the effective size of the data set and essentially forcing us to use only the simplest models. More complicated models suffer from the "curse of dimensionality," which we will explain below.

(2) For years the regression literature has been filled with informal confirmation of the "parsimony principle," which states that simplest models are preferred when possible because no data follows any model exactly, and model departures can be more severe when the training data is overfit by using an overly complex model. Fortunately, we have a straightforward way to guide us toward the proper degree of model complexity: use the model to forecast a held-out (not used for training) testing set and accept the model that performs best on the testing set. A complete treatment of this issue uses what is known as cross-validation to repeatedly divide the data into training and testing sets. Cross-validation is very important when the data sets are small. In our case, we simply did a one-time division into training and testing sets because the data sets were reasonably large.

4.2.1 Linear AR models

Our simplest model assumes that f is linear in this sense:

$$f(x_1, x_2, \ldots, x_p) = a_0 + \sum_{j=1}^{p} a_j x_j \,. \tag{4}$$

For notational convenience we have dropped the distinction among the three time series X, Y, and Z and have lumped all candidate predictors together to form p predictors. An important issue is how to choose the lag for each of the time series. We always use a "trial-and-error" approach starting with lag 1 only for the X series (the series to be predicted) and starting with lag 0 for the

other series. The winning method minimizes the sum of squared errors on a held-out testing set. If the predictors are raised to powers other than 1, we would call it a polynomial regression. Four of our 12 methods are linear regression methods: one is the traditional ordinary least squares (ols) approach that minimizes the sum of squared errors to fit parameters $a_0, a_1, \ldots, a_p$. A second method minimizes the median (lms) of the squared residuals attempting to reduce sensitivity to outlying observations. The third method is a robust regression (rreg) that uses a more general "M-estimation" procedure that uses iteratively re-weighted least squares. The second and third methods both attempt to reduce sensitivity to outlying observations, which are known to cause problems with ordinary least squares. The fourth method is a class of the generalized linear model (glm) that allows the user to specify the error distribution F and thereby use parameter estimation that is designed to be optimal for that distribution. We assumed that the error variance was approximately Poisson-distributed for this method. In fact, we know of some extreme departure from Poisson variance, so we had little hope for this method. In all four cases, the forecast errors could be analyzed to suggest model departure, such as nonlinearity. If nonlinearity is detected, then a good procedure is to search for a suitable transform for some of the predictors or to add polynomial terms in the predictors. We have not detected nonlinearity in the forecast errors for most of the data, so we suspect that the linear models will suffice. However, we are assembling a toolkit of model-fitting methods because they might be needed for other data sets. In that spirit, the next group of methods makes less restrictive assumptions about f. Also included in this first group is the exponential smoother, which uses a weighted average (exponentially decaying weights) of the most recent observations to forecast the present observation. This method is equivalent to a linear (MA) model fit to the first differences of a series, in the case we consider here. The linear (MA) model fitted to the first differences of a series is denoted ARIMA(0,1,1) to denote that first differences are analyzed (original data is nonstationary) and that MA with lag 1 is used. For the exponential smoother [equivalently, the ARIMA(0,1,1)] we use only prior gammas to forecast the present gamma counts. If we used the electron and proton series, we would have to adapt the exponential smoother more than we considered appropriate at this time.

4.2.2 Nonlinear AR models

The second group of methods consists of the generalized additive model (gam), MARS (multivariate adaptive regression with splines), and projection pursuit regression (ppreg). All of these methods restrict the functional form for f somewhat, but not nearly as much as the linear methods.

1) *The gam model.* This model assumes

$$f(x_1, x_2, \ldots, x_p) = \sum_{j=1}^{p} f_j(x_j). \tag{5}$$

Certainly (5) generalizes (4), as the individual f_j are arbitrary smooth functions. Model (5) is called additive to emphasize that none of the p predictors "interact," so for example, there is

no term such as $f_{12}(x_1x_1)$. If such interaction terms are considered necessary, then they must be put in "by hand" by defining a new variable $x_{p+1} = x_1x_1$. The individual f_j are estimated by a "curve smoother" that can be described qualitatively with the help of Fig. 2. Figure 2 illustrates a "curve smoother" that the human eye could do quite well. Perhaps surprisingly, training software to fit the smoother is non-trivial, and we discuss it further in Reference 8. For now, accept that estimating a smooth function using a "curve smoother" algorithm is feasible in one-dimension or perhaps in a few dimensions. Occasionally, if enough data follows Equation (2), then a multidimensional curve smoother is feasible. However, the idea behind the gam model in (5) is that the "curse of dimensionality" can be mitigated by restricting the functional form to be additive.

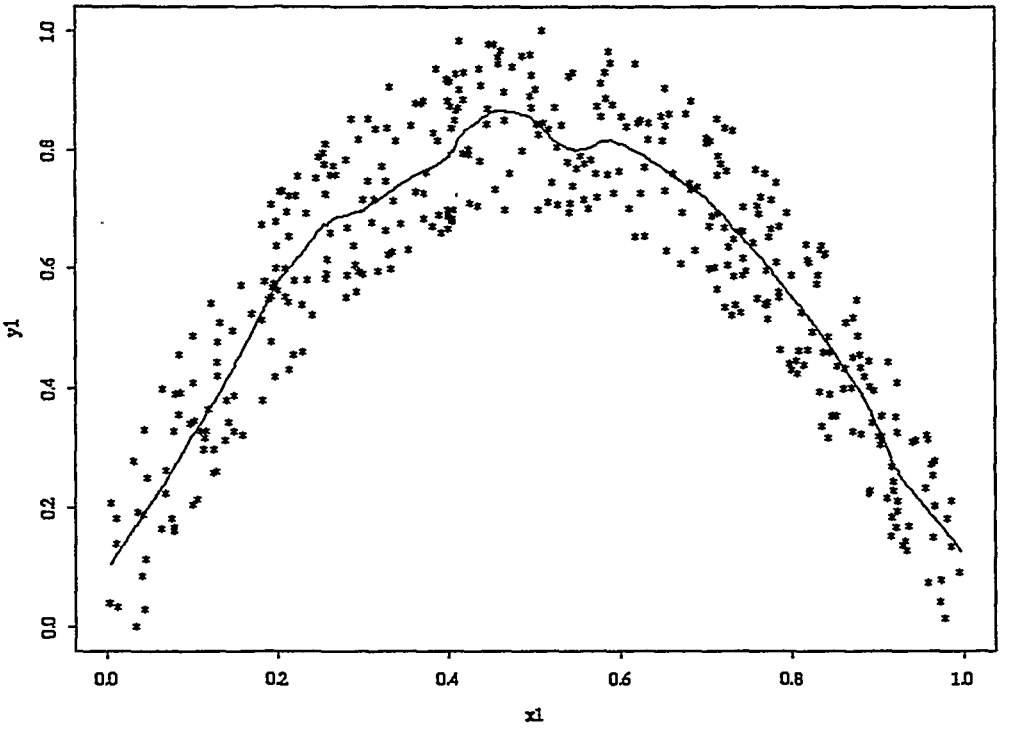

FIGURE 2. Scatterplot illustration of a one-dimensional curve smoother.

2) *MARS.* J. Friedman's MARS method is perhaps one of the most important advances in applied statistics in the last 10 years. MARS is an extension of CART (classification and regression trees, which we describe in more detail in section 4) that addresses the weakness of CART for regression. This weakness can be explained as follows. With CART, the predictor space is split into non-overlapping regions and the predicted response is the average of the response for cases with predictors that fell into that region of the predictor space. This procedure can lead to a discontinuous response surface, which is nearly never desirable. The MARS methodology removed the discontinuity of the response surface by effectively allowing overlap between the regions. See Ref. 9 for more detail. The MARS software is public domain, in

several forms. We have experimented with a Fortran version and a version compiled for S+. Results in this report refer to the S+ version contributed by Trevor Hastie.

3) *Projection Pursuit Regression.* The idea in ppreg is to seek M new direction vectors $\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_M$ and good nonlinear transformations $\phi_1, \phi_2, \ldots, \phi_M$ so that

$$f(x_1, x_2, \ldots, x_p) \approx \sum_{j=1}^{M} \phi_j\left(\tilde{a}_j^T \tilde{x}\right) \tag{6}$$

because the inner product operation $\tilde{a}_j^T \tilde{x}$ represents a projection of $\tilde{x}$ onto the direction $\tilde{a}_j$.

4.2.3 Nonparametric AR models

The last group of models includes local regression, fuzzy controller, and nonparametric smoothing.

1) *Local regression (loess in S+).* We illustrated a statistically motivated curve smoother in Fig. 2. In our view, a local linear or local polynomial model is a special case of a statistically motivated curve smoother.

2) *Nonparametric smoothing (written for KF in 1994).* We will present the equations for our one dimensional curve smoother. The higher dimensional smoother is a natural extension. Our curve smoother is similar conceptually to other statistically motivated curve smoothers.

$$\hat{f}(x) = \left(\left[(1/(n-1)) \sum_{j=1}^{n-1} X_{j+1} w\left(\frac{x - X_j}{h}\right) \right] / \left[(1/n) \sum_{j=1}^{n} w\left(\frac{x - X_j}{h}\right) \right] \right). \tag{7}$$

Note the "hat" notation in $\hat{f}$ which means that the rhs is an estimate of the true f. This reflects a change from all previously described models because we do not introduce an assumed restricted form for f. Therefore, we simply present an intuitive way to estimate f at the point x as follows. All data contributes to the estimated value via a weighted average, with weight given by the distance of the data points from x. A data point, say X_k, that is far from x simply won't contribute much to the estimate at x, provided we choose the smoothing parameter h and the weight function w so that $w\left(\frac{x - X_j}{h}\right)$, which is the weighting term, is small when $x - X_j$ is large. Consider Fig. 2 again, in the light of this description. For a given value $X = x$, the estimate for Y is primarily determined by those Y values that correspond to X values near x. This is a simple idea, but selecting the smoothing degree remains somewhat of an art despite attempts to automate the choice of bandwidth h. However, by using held-out testing sets, it is possible to do a reasonable job of automating the choice of h. Experience and theory suggest that the choice of h is more critical than the choice of w. Typically simple smooth functions such as a Gaussian-shaped function e^{-x^2} are a good choice. Theoretically optimal weight functions such as the Epanechnikov kernel are sometimes suitable, but the best theoretical

shape for the weight function depends on the true function f, so we are not fond of using specially motivated weight functions. In fact, we nearly always use simple Gaussian weight functions and concern ourselves with searching for good h. We give more details in Ref. 8, but the basic idea is to balance the trade-off between bias and variance: too little smoothing overfits the data, reducing bias but increasing variance, and the reverse occurs for too much smoothing.

3) *Fuzzy forecaster (written for KF in 1995).* The fuzzy forecaster has a few forms. The form we use here is often called a fuzzy controller (Ref. 10). In our view, the fuzzy logic approaches to modeling and forecasting time series could benefit from a more statistical approach, especially in the area of choosing the number of fuzzy regions, which is comparable to our choice of bandwidth in Equation (7).

In Ref. 8 we provide more detail on this last group of models, including comparison on five real and five simulated data sets. The conclusion in Ref. 3 is that the fuzzy forecaster offers no advantage over statistically motivated curve smoothers. In fact, our view is that fuzzy logic offers no advantage over statistically motivated approaches in **any** area in which the data is numeric. Fuzzy logic was developed for situations in which numeric data was not available and probably is best applied only in those unfortunate situations.

4.2.4 Summary of the vector AR models considered

First, we are considering vector AR models only because we have relaxed the linearity assumption. Although it may be possible (see Ref. 11 for an ad hoc "two-step" procedure that is under investigation) to treat nonlinear MA models, we have not attempted that with the nudet-detection-related data. Once we restrict attention to AR models, we get access to a host of regression techniques that can be applied as if the data were in the usual regression setting: observe independent cases of data "pairs" $(\tilde{x},y)$ where the $\tilde{x}$ vector is a p-component predictor vector. The only difference in our setting is that successive cases are not independent because of the serial correlation. However, asymptotically (as the number of cases increases) this serial correlation can be ignored for estimating the function. (See Ref. 12).

Most real time series do not exactly follow *any* model, linear or nonlinear. The challenge in such cases is to select a reasonably simple model that captures the relevant behavior of the series. And many time series cannot be made stationary (de-trended) for extremely long time periods. However, many real time series change slowly enough that under their "usual behavior" some kind of model is in effect locally. For example, a particular ARIMA model might be a reasonable model for the first 1000 observations, but a different ARIMA model might be a better model for the next 1000 observations. This complicates the situation, because in that case, we do not care about *all* changes that might occur in a time series model. As another example, a series might follow something like a simple ARIMA(0,1) model [also known as an MA(1) model] but with varying error

variance. This could be due to a probabilistic mechanism that causes the mean to affect the variance. Simply put, large numbers tend to vary more than small numbers, so the error variance might depend on the mean of the series. Depending on the particular application, such a model change might not be of interest. For example, in any nudet-detection project, it is likely that the time series generated by radiation detectors counting the background will have some Poisson-type variance component. The Poisson distribution often arises in particle-counting statistics and the Poisson variance equals its mean. Therefore, we should not be surprised to see higher variability in sections of the time series where the mean is higher.

5. Further Details About "Event Records"

Most of our 1995 effort has been with the "event records" for which we have more frequent observations of the gammas and for a series which we call flld that is proportional to the sum of the protons and electrons. One lesson learned during 1994 was that the anticipated relation between charged particles and gammas had to be carefully defined. Over some time scales, there is an obvious strong relation between, say, the electrons and gammas. For example, in Fig. 3 we show a section of time with the electrons in the top plot and gammas in the bottom plot.

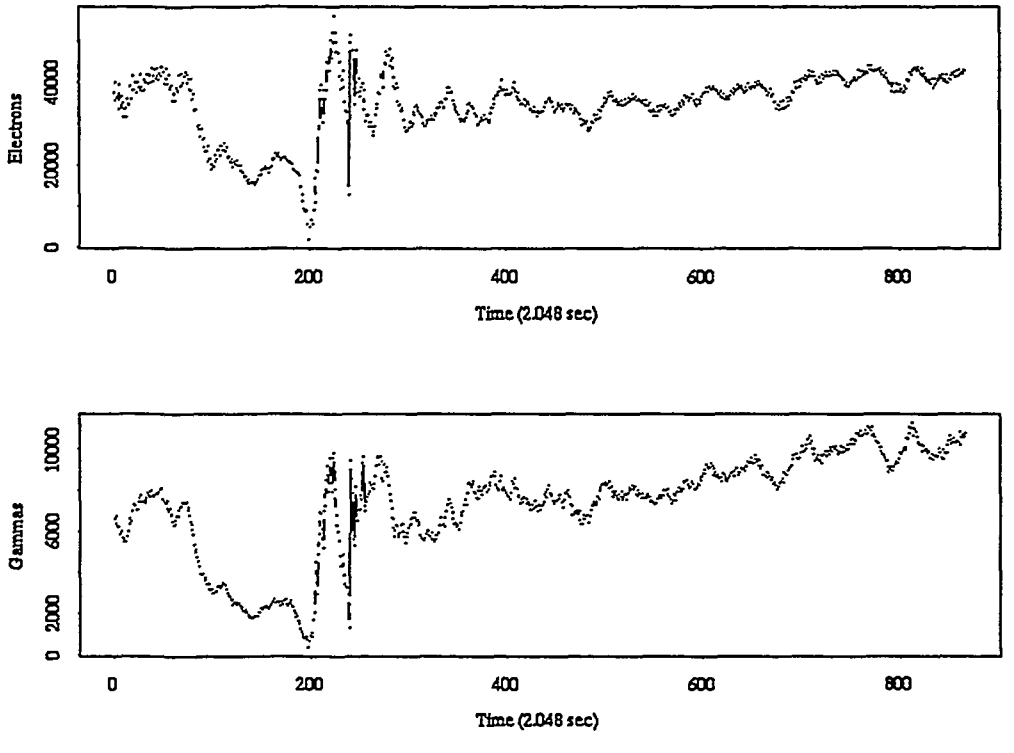

FIGURE 3. Electron counts and gamma counts.

From this plot it appears that it will be simple to use the electrons to forecast the gammas because the electrons appear to be proportional to the gammas. In fact, forecasting the gammas using electrons with data recorded about every 8 sec works fairly well. However, in Fig. 4 we plot the ratio of the electrons to the gammas versus time, and we see that in the region of large variability in the gamma counts, there will also be large variability in the forecast errors using a linear fit of the gamma counts regressed on electron counts.

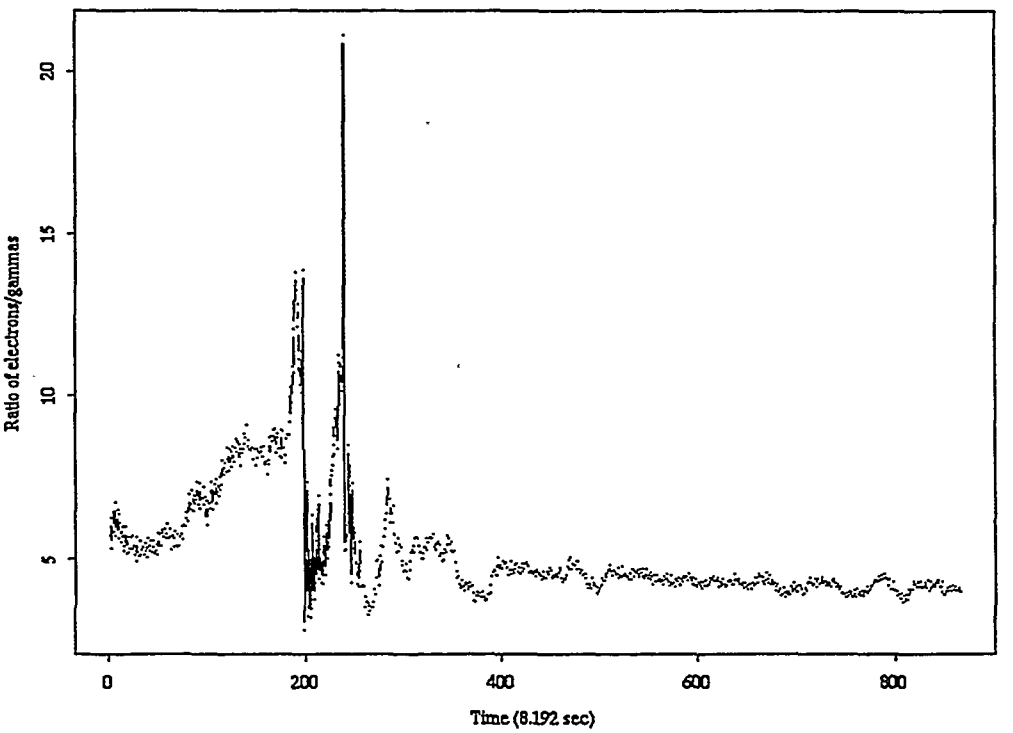

FIGURE 4. Ratio of electrons to gammas over the same time period as shown in Fig. 3.

Forecast errors from a linear fit of the gamma counts to the electron counts are shown in Fig. 5, and we informally see that there will be at least one alarm near observation 220.

Also, our nudet signal is mostly contained within about 2 sec so it is better to use more frequent data. That is a second reason we considered the "event records" in 1995. These event records record data more frequently than every 8 sec and have a "pre-event" buffer that we will assume represents ordinary background. Therefore, using the event records we can (1) check the relation between gammas and charged particles over shorter-than-8-sec time scales and (2) have a better "signal-to-noise" ratio for nudet detection.

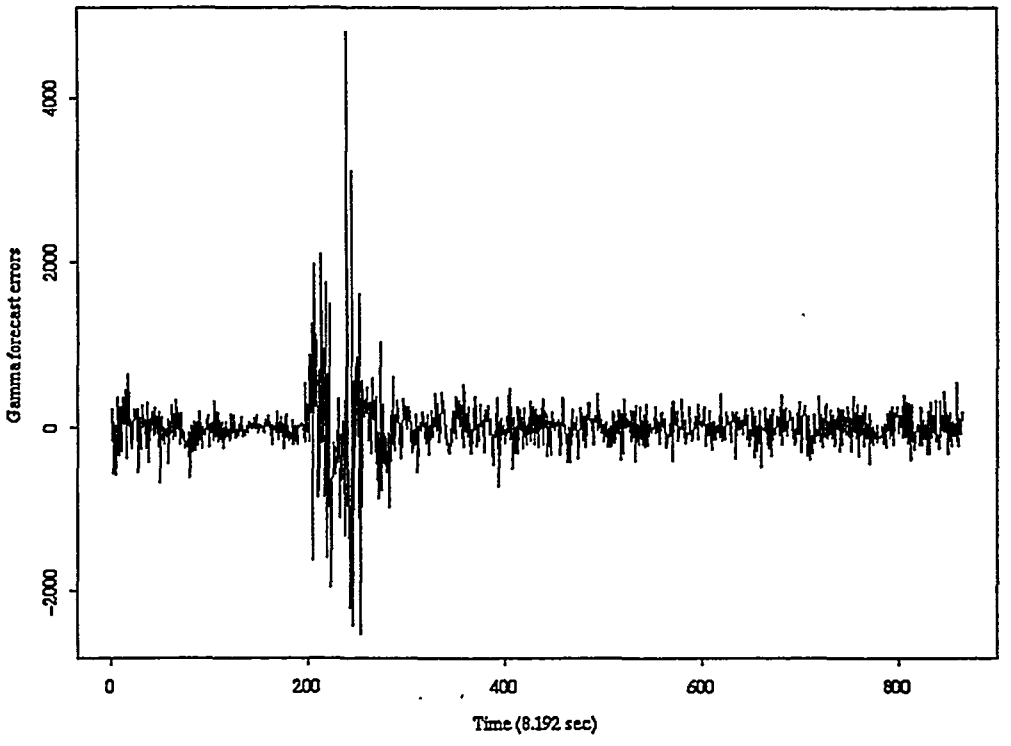

FIGURE 5. The forecast errors that result from a linear model relating gamma counts to electron counts for the region shown in Figs. 3 and 4.

To complete this discussion, we report results of our suite of 11 candidate forecast methods to the section of 878 pairs of concurrent electron and gamma counts. Because this was data with 8.192-sec resolution, 878 pairs represent about 2 hr of data. Here are the two ground rules:

(1) Use the first 2/3 of the data (the first 586 data pairs) to build the model. Then report the average squared forecast error over the last 1/3 of the data (the last 292 data pairs).

(2) Always test (formally with statistical tests or informally with graphical tests) for serial correlation in the forecast errors. If the forecast errors exhibit serial correlation, then the model is not yet acceptable because that serial correlation should be exploited in the model. Forecast-error variances are reported only if the errors exhibit no serial correlation. A more complete model assessment could attempt to check for any serial structure in the forecast errors (not restricted to serial correlation, which only measures the linear structure). We have not attempted to check for nonlinear structure in the forecast errors.

The goal now is to minimize the average squared forecast error of the gammas. We begin with the exponential smoother and ARIMA(0,1,1). These two are theoretically equivalent, but there are slight implementation differences between our exponential smoother and the S+ implementation of ARIMA(0,1,1). The variance of the last 1/3 of the original gamma series is 82,563.

In these first two methods in Table 1, only prior gammas are used to forecast the present gammas. In methods 3 to 12, we use present electrons and protons to forecast the present gammas. The main observation is that none of the methods reduce the forecast-error variance noticeably more than the exponential smoother result. Each of the 11 methods below can be appropriate for a particular kind of time series. In a later report we will document the performance of these methods on other data sets.

TABLE 1. Comparison of the 12 Methods Applied to Forecasting the Gammas

Method	Average Squared Error
GROUP 1:	
Method 1: exponential smoother	27,132
Method 2: ARIMA(0,1,1)	27,256
Method 3: least squares linear model	29,506
Method 4: least median squares linear model	68,770
Method 5: robust regression	29,517
Method 6: general linear model	32,522
GROUP 2:	
Method 7: projection pursuit regression	118,797
Method 8: generalized additive model	44,144
Method 9: MARS (multivariate regression with splines)	37,366
GROUP 3:	
Method 10: loess - a local regression	49,776
Method 11: nonparametric smoothing (written for KF)	36,036
Method 12: fuzzy controller (written for KF)	******

For method 12, the fuzzy controller, we did not include the result because it was applied with a slightly different protocol than we have described. Rather than divide the data into the first 2/3 to train and the last 1/3 to test, we randomly selected 1/2 to train and 1/2 to test. Therefore, results are not directly comparable. We plan to add a version of our fuzzy controller that will divide the

data deterministically into a first 2/3 to train and a last 1/3 to test, but for now we can at least compare the fuzzy controller result to a version of method 10 (loess) that used the same protocol as the fuzzy controller. The resulting average squared forecast errors were 41,143 (fuzzy controller) and 41,699 (loess). Because we repeated the experiment several times, we could assign a confidence interval to both results. We concluded that the two methods do not produce statistically significant differences. We give more detail in Ref. 8.

The main conclusion is that the first group (linear methods) does as well as the more complicated methods. Also, we are puzzled by the poor performance of projection pursuit regression, especially because it performed competitively on the training data, which was the first 2/3 of the data set. We plan to experiment further with projection pursuit because we have been told that it sometimes gives the best results. Our conclusion with this particular data set, however, is that the data do not follow any model very well, so the simplest model assumptions are the best.

6. Pattern Recognition to Separate False Alarms from True Events

In section 4 we describe two related issues: preparation of the "event records" for analysis by pattern recognition methods and development, application, and adaptation of several pattern recognition methods.

6.1 Preparing the Event Records for Analysis by Pattern Recognition Methods

The event records contained more information than we needed, so we restricted attention to only the flld (proportional to the sum of electrons and protons) and gamma counts. For both the flld counts and gamma counts, there are about 20 counts of "pre-event" data, 5 counts of "event" data, and 13 counts of "post-event" data. The post-event data is recorded less frequently that the pre-event and event data. In Fig. 6 we show 4 plots: 6a and 6b show the gammas (flld) from a randomly selected event record, 6c shows gammas versus flld for the same event record, and 6d shows gammas versus flld for two other randomly selected event records. The disappointing message is that over these time scales we do not see much relation between the flld and the gamma counts.

A separate report (Ref. 7) gives more detail about these event records, including a model-based summary of the relation between the flld and gamma counts. By "model-based" summary, we mean, for example, that a linear model was fit relating the present gamma count to the present flld count and that the estimated linear coefficient was stored in a file to be summarized. Many of the linear coefficients were actually negative. And there is strong periodicity in the flld counts and negligible periodicity in the gamma counts. That fact alone suggests that the flld counts cannot be terribly well related to the gamma counts. The weak relation between the flld counts and the gamma counts makes it difficult to predict whether it is better to forecast the gamma series using prior gammas or some combination of prior gammas and flld. See Ref. 7 for more detail.

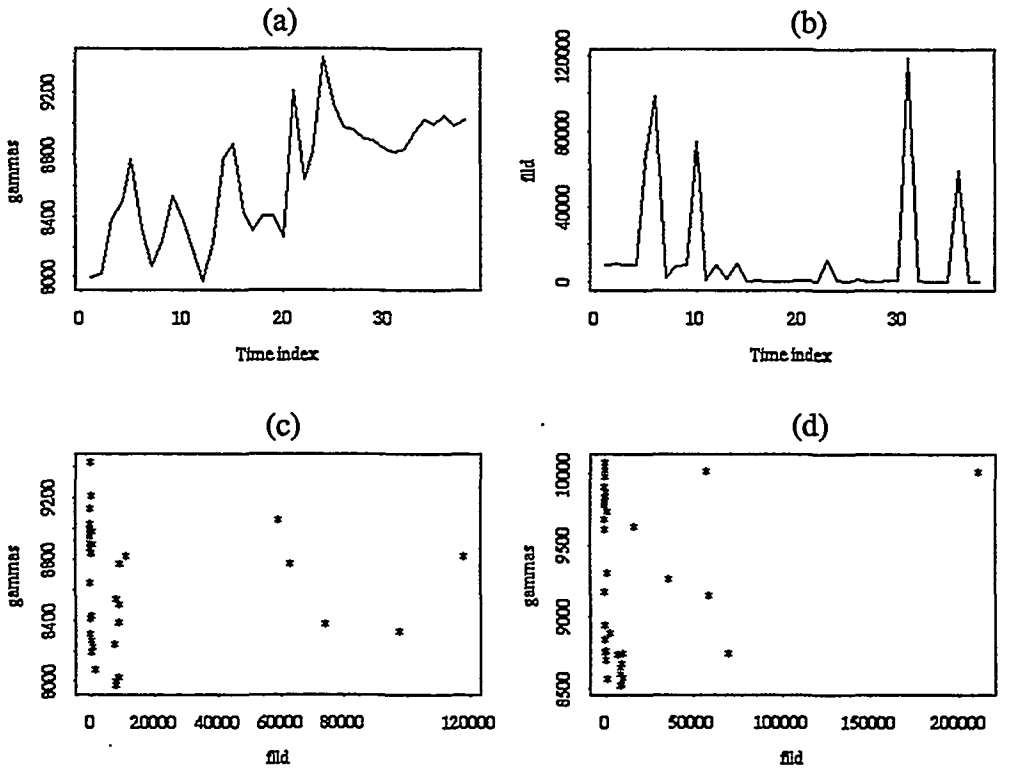

FIGURE 6. Example event records. (a) gammas versus time index (38 points), (b) filld versus time index, (c) gammas versus filld for same record as in (a) and (b), and (d) gammas versus filld for another randomly selected record.

To this point we have mentioned 76 candidate features (38 gamma counts plus 38 filld counts) to attempt to discriminate nudets from unusual background. There are two other features: the data-gathering hardware system on a particular satellite (coded as system 1 or system 2) and the satellite identification (coded as 1, 2, or 3). We also "created" the following 45 additional candidate features in groups (1) - (12) below.

1. Exponential smoother results were obtained by applying the exponential smoother to the first 20 points of the gamma series and searching for and recording the smoothing parameter that minimizes forecast error variance, the minimized variance, and the ratio of the minimized variance to the variance of the first 20 points of the gamma series. (3 features)

2. Poisson checks were obtained by binning the data into five regions: points 1-10, points 11-20, points 21-25, points 26-28, and points 29-38. The reasons for doing so are as follows: (1) many of the event records exhibit reasonably slow variation, so the gamma counts for those records should be approximately Poisson-distributed over fairly short sections of time; (2) the event records were supposedly triggered by point number 21, so points 21-25 should be treated separately from points 1-20; and (3) the post-event points have two levels of resolution, one for points 26-28 and the other for points 29-38. We account for the three levels of resolution by rescaling, computing the rescaled variance-to-mean ratio for each of the five regions, and thereby creating five features. (5 features)

3. The maximum forecast error over points 16 to 25 was recorded using a simple moving (unweighted moving average) average, the index in 16-25 for which the maximum error occurs, and the postavg-preavg (post and pre are defined relative to the index where the maximum error occurs). The maximum forecast error is recorded as a ratio of the actual error to the square root of the relevant moving average. (3 features)

4. Mean and variance of gammas were recorded over each of four regions: 1-20, 21-25, 26-28, and 29-38. (8 features)

5. Mean, variance, and slope of gammas were recorded over points 1-17. (3 features)

6. Mean of gammas were recorded over points 18-20. (1 feature)

7. The index i after index 21 for which the 3-point average (indices i-2, i-1, i) falls below the gamma count at index 21 defines a peak "width." The following features were created: peak (among points 21-38) relative height (absolute height minus 3-point average in (6) above), peak absolute height, and width. (3 features)

8. A measure of overall trend in the 38 gamma counts was the difference between the gamma means over 29-38 and over 1-20. (1 feature)

9. Based on observing that some of the calibration events showed a large gamma count at index 1 followed by a small gamma count, we considered the gamma count at index 2 minus the gamma count at index 1 to be a feature. (1 feature)

10. We fit a linear model to the first 15 gammas as a function of the first 15 flld counts. We used the linear coefficient (slope of gammas versus flld) to forecast the next five gammas (points 16-20) and recorded the corresponding five forecast errors, their variance and mean, and the linear coefficient. (8 features)

11. Same as (10) above, but we used the first 20 gammas and the first 20 flld counts to estimate the linear coefficient and forecast the next five gammas (points 21-25). We recorded the corresponding five forecast errors, their variance, and mean. (7 features)

12. We fit a linear model to the gammas versus time over points 1-20 and 29-38. We recorded the linear coefficients. (2 features)

In a few preliminary experiments, we included the 38 flld counts among the long list of candidate features. However, because the gamma series is of primary interest and the flld series is used to help create several of the candidate features, we did not include the 38 flld counts themselves among the candidate features in the results reported here. The number of candidate features is then reduced to 38 (gamma counts) plus 2 (system type and satellite) plus 45 (45 derived features described in the 12 groups above) equals 85.

For completeness, we also mention here that because we had access to quite complete event records, we could compute user-defined gamma counts by summing desired channels. This was accomplished by a pearl script and allowed us to completely eliminate one category of events. The eliminated category is referred to as operator errors, which are created by summing other-than-the-nominal energy channels for the gamma counters. We summed the nominal energy channels to eliminate the operator-error category. We were left with three categories: "true," "cal1," and "cal2." We think of the trues as candidates to be nudets because there is no assignable cause for the rapid gamma fluctuation, whereas with the calibration events of type cal1 or cal2, there is an assignable cause for the rapid gamma fluctuation. Using all 85 candidate features, we created the following file:

1) file3.df — 85 features, 3 classes (true – 194 cases, cal1 – 50 cases, cal2 – 19 cases). There were a total of 263 cases.

We also wanted to experiment with avoiding any feature that directly related to the magnitude of the gamma counts because we knew that there was considerable variability in the gamma background. Therefore, we eliminated the 38 gamma counts and any feature that directly related to the gamma count to create the following file:

2) file4.df — 33 features, 3 classes — same as file3.df, but only uses relative height of gamma peak, width of gamma peak, gamma count at index2 minus gamma count at index1, the 8 features involving forecast errors in group 10, the 7 features involving forecast errors in group 11, the 2 features involving linear trends over points 1-20 and over points 29-38, the 11 features described in groups (1) - (3), and system type and satellite.

The next file (file5.df) uses the same candidate features as file3.df but adds a nudet class. We obtained the nudet cases from a random sample of event records, which did not satisfy our alarm criterion. Those records were judged to be reasonably representative of ordinary background, so the simulated nudets were added to points 21-25 in those records. The idea is that we did not want to consider the case where a nudet occurred during a section of unusual background. The simulated nudet added counts from a Poisson distribution with an average shape distributed over points 21-25, which was determined by repeated execution of the nudet simulation code that was written during the 1994 KF project.

3) file5.df — 85 features, 4 classes — same as file3.df, but we added the nudet class for a total of 273 cases.

The last file (file6.df) uses the same candidate features as file4.df, but as with file5.df, a nudet class is added.

4) file6.df — 33 features, 4 classes — same as file4.df, but we added the nudet class for a total of 273 cases.

The file names are unimportant here except that we refer to them in several of the plots in section 5.2.

The main reason for considering file3.df and file4.df, which do not have nudet classes, is to apply cluster analysis to determine whether the true class seems to have distinct mechanisms giving rise to the erratic counts. Also, three of the pattern recognition techniques essentially apply a form of cluster analysis separately to each class to see how many clusters are present in each class and then use some representation of the cluster center to discriminate among cases. More detail is given in section 5.3. In section 5.2 we present some of the results of cluster analysis applied to file3.df and file4.df.

6.2 Results of Cluster Analysis

We applied three clustering techniques to the class=true cases from file3.df and file4.df: kmeans, hierarchical clustering, and model-based clustering. We give a brief description of each below.

1) kmeans is one of the oldest clustering methods. If $k = 2$, then the algorithm searches for the best partition of the cases into two clusters such that the within-cluster variance is small compared to the between-cluster variance. We apply k-means for a range of k from 2 to 10 and select the value of k using a criterion suggested by Hartigan (Ref. 13). The criterion accepts adding a cluster to increase from k to $k + 1$ clusters if the within-cluster sum of squares is sufficiently reduced. Applied to file3.df, there is reasonably convincing evidence for four clusters.

2) Hierarchical clustering is the simplest clustering method to describe. Compute the distance between each pair of cases. Group the closest two cases first, then add to that cluster the case that is closest to that first cluster. There are several varieties of hierarchical clustering depending on how distances from a case to a cluster or from a cluster to a cluster are determined. A common choice is to use the average distance between a given case and each case in a cluster as the distance from a case to a cluster. Another choice is to use the largest distance between a given case and each case in a cluster. In Fig. 7 we show the results of a hierarchical clustering using the latter definition of distance. The main features in Fig. 7 are that case 63 is an outlier and that there is informal evidence for three other clusters. The informal evidence is the same as that used from the kmeans criterion: the within-cluster variance is reasonably well reduced by choosing three main clusters and one outlying cluster of size one. So, as with kmeans, we again find reasonably convincing evidence for four clusters.

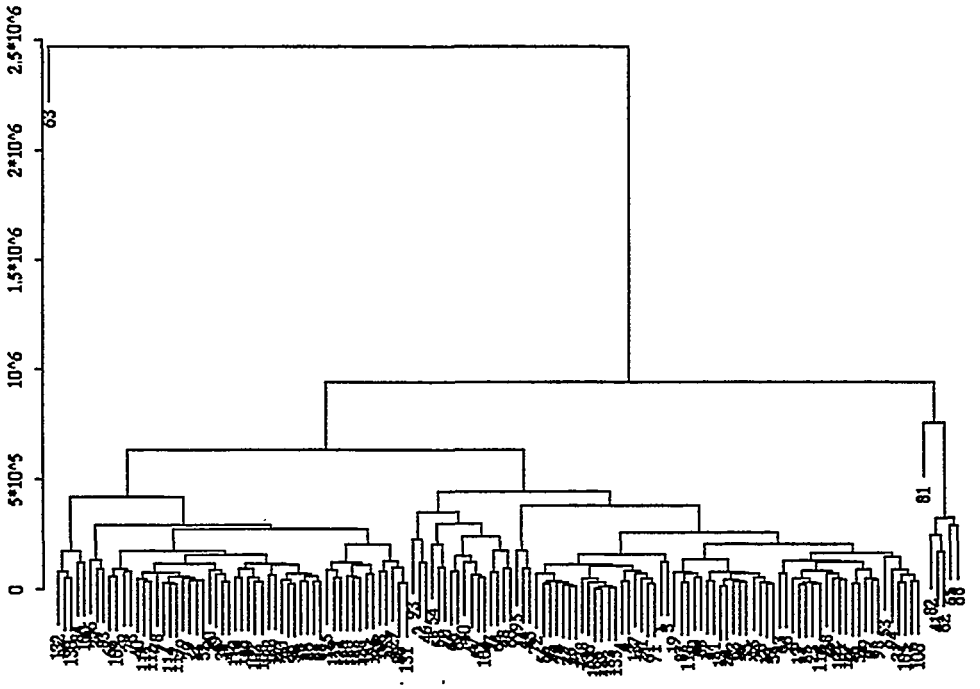

FIGURE 7. Result of hierarchical clustering applied to the class=true cases from file3.df.

Because the hierarchical clustering indicated that case 63 is an outlier, we investigated the cause. First, we used a standard dimension-reduction technique that uses the covariance matrix containing the variances of each feature on the diagonal of the matrix and the covariances between each feature in the off-diagonals. The dimension-reduction technique then uses eigenvectors of the covariance matrix to define a new coordinate system for the data. The first coordinate (called the first principal coordinate) is the eigenvector corresponding to the largest eigenvalue, the second coordinate corresponds to the second largest eigenvalue, etc. If the data is projected in the direction of the principal components (PC), then the variance of transformed data for PC1 is maximum, and the variance of PC2 is next largest constrained to be uncorrelated with PC1. In Fig. 8 we show the fraction of the total variance of the original data (sum of diagonal entries of covariance matrix) by each PC. Informally, we see that the first two or perhaps the first three PCs contain most of the variance. Therefore, we can see which variables (features) from case 63 contribute most to the first two or three PCs. Also, we can plot PC1 versus PC2 for each case and confirm that case 63 still appears to be an outlier even for this reduced-representation of the data (see Fig. 9). By calculating the contribution of each variable to PC1 and PC2, we can see that case 63 is an outlier for a surprising reason: the variance of the gamma counts over points 1-20 is unusually high while the mean over points 1-20 is unusually low. To further investigate case 63, we rescaled the data so that all variances would equal one. It is a well-known fact that clustering results using covariance matrices can be very different from results using correlation matrices, so we wanted to know if case 63

would still appear to be an outlier using the rescaled data. The result was that case 63 continues to appear as an outlier on the rescaled data, but the important variables become the Poisson checks rather than the variance and mean gamma counts over points 1-20. We currently have no explanation for this behavior.

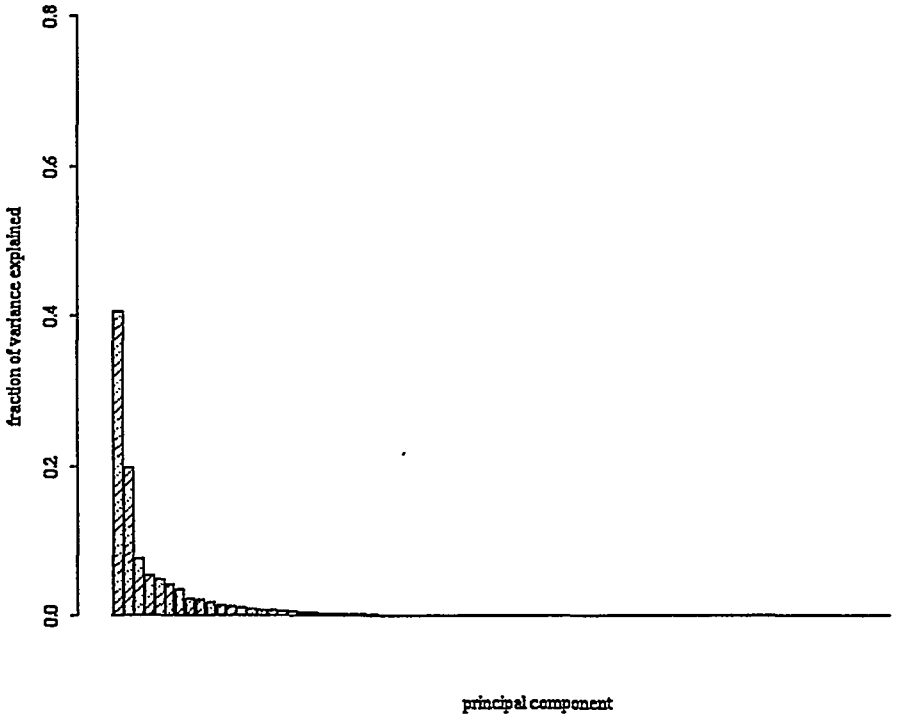

FIGURE 8. Plot of principal components for file3.df.

3) Model-based clustering is a relatively new clustering technique that uses an extended version of hierarchical clustering (Ref. 14). The extensions are that a Bayesian criterion helps choose the number of clusters and noise or outliers can be explicitly modeled. We experienced out-of-memory problems with our system using model-based clustering on file3.df, but applying it to file4.df (fewer features), we again found evidence for four clusters based on the Bayesian criterion.

We have also applied the three clustering methods to file4.df and again found reasonable evidence for three clusters using kmeans and hierarchical clustering, but model-based clustering did not strongly suggest a best number of clusters. Also, hierarchical clustering suggested case 63 to be an outlier, and when we removed case 63, case 81 appeared to be an outlier. We plan to investigate possible reasons for this behavior.

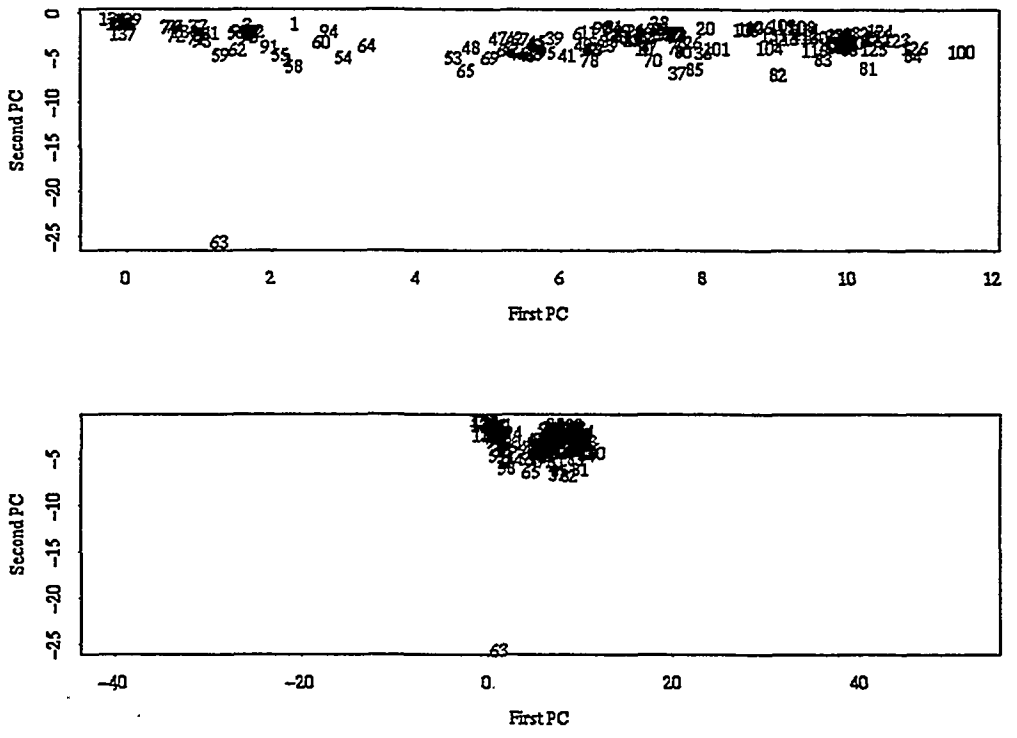

FIGURE 9. Plot of PC2 versus PC1 for file3.df. Bottom plot forces equal scales on vertical and horizontal axes; top plot does not.

We summarize as follows. One of the useful conclusions from the cluster analysis was that one or two true outliers were identified, and a second useful conclusion was a reasonably strong suggestion that the class=true cases contain three clusters using either the full set of features (file3.df) or the reduced set of features (file4.df).

6.3 Adaptation and Application of Several Pattern Recognition Methods

In this section we introduce and describe the following pattern recognition methods: decision trees, linear discriminant analysis, mixture discriminant analysis, flexible discriminant analysis, k-nearest-neighbor methods, a new mixture discriminant analysis (written for KF), and a particular (well-known) neural network called learning vector quantization.

6.3.1 Decision trees

There are several approaches to building decision trees. We give an example tree using file6.df (4 classes, 33 candidate features) in Fig. 10. The approach we prefer is that implemented in the commercial software CART (classification and regression trees). We mentioned CART in section 3.2.2 in connection with MARS. The key issue is how to select one tree from the infinitely many possible trees or how to combine information from several selected trees. We postpone until 1996 the issue of combining information from multiple trees. Here we present single trees that were selected from a combination of ad hoc and formal methods. The idea behind CART trees is to

build a large tree with many terminal nodes, then to use held-out test data to help select the degree to which the large tree should be pruned back. As is always the case, there is a tendency to overfit the training data (build too many terminal nodes), so it is essential to use held-out test data to counter this tendency. Non-terminal nodes are associated with a split criterion such as file6.30 < 30.574 at the root node in Fig. 10. The notation conveys the idea that feature (or predictor) variable number 30 was used and cases with variable 30 < 30.574 travel left through the tree. Variable 30 is the maximum forecast error between points 16 and 25. Such a split criterion is obtained by a trial-and-error method that tries splitting a given node at all feasible breakpoints for numeric variables, or for all possible subset partitions for categorical variables. Other important variables (used in splitting criteria) in the tree shown in Fig. 10 are as follows: variable 31 is the position of the maximum error, variable 12 is the slope of the gamma counts over points 1-20, and variable 13 is the error at point 21. Notice that the file 30.26 < 4.05 node splits into two nodes that are both labeled class 1. Those labels are the predicted class, so splitting a node into two nodes that will both be predicted to be class1 cases has no effect on the performance of this tree. However, because the tree was first grown larger and then pruned back, it did have an effect on the larger trees.

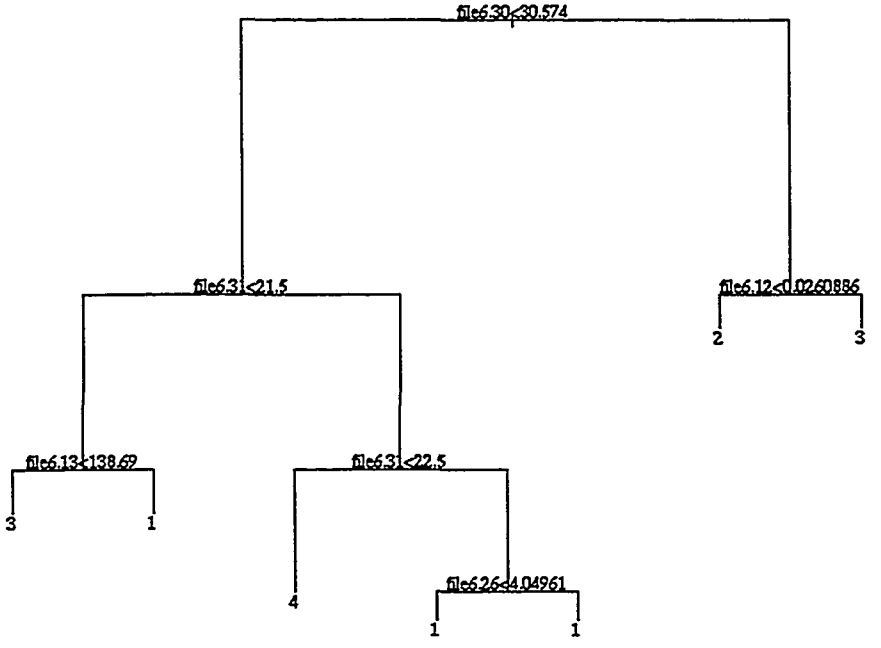

FIGURE 10. Decision tree for file with 4 classes and 33 candidate features.

The CART software also considers using linear combinations of variables at a given node, but we rarely find that linear combinations of variables are needed. Therefore, this report restricts attention to results using CART-like software that is in the statistical programming language S+, which does not consider linear combinations of variables as possible split criteria. The advantages of

decision trees include the following: robustness with respect to outliers, easy allowance for missing data through use of surrogate splits (each node has second, third, ..., best split criteria that can be used if a case is missing the variable used by the best split criterion), excellent data exploration capabilities, and simplicity of use once constructed. For further details, see Ref. 15.

In Fig. 11 we show a decision tree for file5.df with 4 classes and 85 candidate features. Variable 82 is the average forecast error over points 21-25. This is a satisfying result because the Neyman-Pearson Lemma would suggest that either a weighted or unweighted sum of the forecast errors over points 21-25 would be a good discriminator. See Ref. 7 for further details on the application of the Neyman-Pearson Lemma to this type of problem. Variable 55 is the average of the gamma counts over points 29-38 minus the average of the gamma counts over points 1 to 20. Variable 44 is the variance of the gamma counts over points 1 to 20. Recall that case 63 was an outlier largely because it had a very high variance and a low mean for the gamma counts over points 1 to 20. Variable 83 is the slope of the gamma counts over points 1 to 20. As in Fig. 10, terminal nodes are labeled with the predicted class for cases that fall into that node.

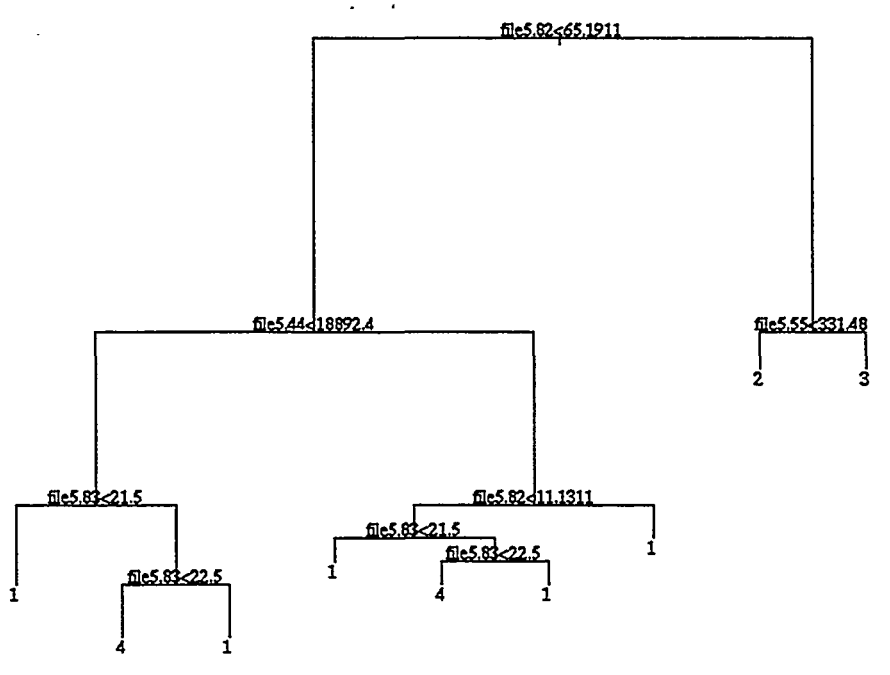

FIGURE 11. Decision tree for file with 4 classes and 85 candidate features.

6.3.2 Linear discriminant analysis (lda)

Linear discriminant analysis is the original (dating to 1930s with R. Fisher) pattern recognition method. The assumptions are as follows: for a given class i, the data has a multivariate normal distribution with mean vector $\tilde{\mu}_i$ and covariance matrix Σ, which is denoted $\tilde{x}|i \sim N(\tilde{\mu}_i, \Sigma)$. That is, the mechanisms generating the data for different classes differ only in the mean vector. Under

that assumption, lda is the theoretical best method and is also called the Bayes method for that reason. Of course, real data never follow any theoretical distribution exactly and, even if the normality assumption was reasonably well met, the assumption of equal covariance matrices remains as well as the assumption that each class has only one mean vector. Nonetheless, lda performs remarkably well for a variety of real data sets; Ida continues to be a benchmark method.

6.3.3 k-nearest neighbor methods (knn)

The knn method is the simplest to describe. Classify a given case in the testing set according to the classes assigned to the nearest (in the predictor space) k cases in the training set by using majority rule. Break ties randomly. For example, with $k = 2$, if the two nearest cases in the training set have class 1 and 2, then assign class 1 with probability 1/2 and class 2 with probability 1/2. The important issue is the choice of metric to compute distances. Also, for extensive use of knn, it is important to restrict the size of the training set because of the requirement to compute and sort all distances between each test case and all training cases.

6.3.4 Mixture discriminant analysis (mda)

Mixture discriminant analysis is a natural extension of lda, although it is very recent (Ref. 16). The idea is to allow any number of mean vectors for a given class. A clever application of the EM algorithm (estimation-maximization) is to treat the issue that the proportion of cases for a given class that belong to a particular subclass is unknown and therefore is treated as missing in the EM algorithm. The number of subclasses for each class is treated as a "meta-parameter," not formally treated in the EM algorithm but treated by trial and error.

6.3.5 Modified mixture discriminant analysis (mmda)

Modified mda was written explicitly for KF in 1995. It is a simpler version of mda that uses kmeans or hierarchical clustering within a given class to suggest the appropriate number of subclasses (clusters) for each class. Recall from section 5.2 that we have had computational difficulties with model-based clustering, so model-based clustering is currently not an option. Having selected a good number of subclass "centers," we apply the knn approach using the subclass centers with class labels to replace the training data used in the knn approach.

6.3.6 Neural network: learning vector quantization (lvq)

The lvq neural network is similar to our mmda, but with the resulting subclass "centers" replaced by "codebook vectors." The codebook vectors are selected by trial and error as well as the number of codebook vectors per class. These vectors are not likely to be either an actual training case or a mean vector for a group of training cases because the iterative trial-and-error method works in a unique way described, for example, in Ref. 17.

6.3.7 Flexible discriminant analysis (fda)

Flexible discriminant analysis is another recent (Ref. 18) method that exploits the following not-well-known fact. Linear discriminant analysis can be derived by repeated linear regression of the class (viewed as a response) on the predictors. In the first regression, all class=1 cases have response=1 and all other cases have response=0. In the second regression, all class=2 cases have response=1 and all other cases have response=0, and so on. The end result will be estimated (scaled) probabilities of each class, which can be used to predict class membership. The idea of fda is to replace the linear restriction with any of the regression methods, such as those described in section 3.2. Our implementation currently uses MARS only.

7. Results of the Seven Pattern Recognition Methods

In this section (Table 2) we give results of each of the seven pattern recognition methods to file5.df, file6.df, and to a simulated data set we call waveform. The waveform data is described in Ref. 16 and is considered to be a challenging pattern recognition problem for which mda is ideally designed. The theoretically lowest possible misclassification probability is .14 for the waveform data. We used 400 cases to train and 400 cases to test the waveform data.

TABLE 2. Comparison of the Seven Pattern-Recognition Methods on Two Real and One Simulated Data Set

Method	Estimated Misclassification Probability		
	file5.df	**file6.df**	**waveform**
tree	.12	.14	.27
lda	.20	.20	.18
knn			
1nn	.20	.19	.25
3nn	.21	.17	.21
5nn	.20	.18	.17
10nn	.20	.19	.18
mda	.18	.22	.16
mmda	.46	.38	.18
fda	.16	.20	.21
lvq	.35	.38	.16

Recall that file5.df has 4 classes, 85 predictors, and 373 cases. File6.df has 4 classes, 33 predictors, and 373 cases. We presented a selected decision tree for file6.df in Fig. 10 and a decision tree for file5.df in Fig. 11. We created the training and testing set as follows. The number of cases for classes 1-4 were 194, 50, 19, and 110, respectively. One half of the cases for each class were randomly selected for training and the other half were used for testing. In Table 2 we record the misclassification rate for the held-out test cases.

Note that for file5.df and file6.df the decision tree performs best, whereas the mmda and lvq perform the worst. Recall that the waveform data was designed to showcase mda so it is not surprising that mda does the best on that data. We were pleased with the performance of mmda on the waveform data, however, and although we are disappointed by the poor performance of mmda on file5.df and file6.df, we do believe the method can be competitive on some data sets. We are also surprised by the poor performance of lvq on file5.df and file6.df. We should emphasize that we did not attempt to fine tune any of the methods except to create file6.df, which contained fewer candidate predictors. The results should be interpreted accordingly. Our intentions in applying and developing pattern recognition methods in such a setting have been to (1) reduce the false alarm rate by sending all alarms into a discriminant function to attempt to separate true from false alarms and (2) provide a means to better understand the background data, as in the present case where there appears to be three distinct clusters in the class=true cases. To formalize the reduction in the false alarm rate, consider the result with the decision tree. The "confusion matrix" in Table 3 below gives the true class in each column and the tree's prediction in each row.

TABLE 3. Confusion Matrix for the Tree Classifier for file6.df

	1	2	3	4
1	93	0	7	9
2	0	23	0	0
3	0	2	3	0
4	4	0	0	46

In Table 3, consider only the cases for which the tree predicted class=4 (nudet) when the class was not 4. That occurred 4 times out of 50. That is, of the 50 times that class=4 was predicted, in only 4 cases was the true class not 4. This is an 8% false alarm rate. Therefore, the overall false alarm rate has been reduced to 8% of the original false alarm rate that was in effect in the rate of creation of the event records. As an important aside, we are beginning to experiment with building multiple trees based on random resamples (bootstrap samples) of the training set and then using majority rule to classify. This method is being developed by Brieman from the CART team (Ref. 15) and is called bagging (bootstrap aggregation). With the waveform data, the misclassification rate

of the bagged tree is about .16, which is competitive with any of the methods. Therefore, we believe that tree methods or bagged-tree methods show great promise for a wide range of data sets.

8. Summary and Conclusions

We have presented a suite of approaches to modeling and forecasting the ordinary background to enhance the method used for flagging unusual background that must be further analyzed by pattern recognition. The suite ranged from the most basic to the most modern and each method will have some suitable domain of application. In this report the best modeling method had the minimum average squared error for the held-out testing set.

Also, we have presented a detailed study of the application of pattern recognition in the setting of event records from a nudet-detection project. We anticipate that **many** surveillance settings will operate in a similar manner: monitor a subset of key time series and keep a short buffer of data available at perhaps higher resolution for all time series that can be further analyzed when the "suspicion level" is raised as judged by some alarm threshold. Follow all alarms with "anomaly resolution" to attempt to assign causes to each alarm. For example, note from the confusion matrix shown in section 6 that all 23 of the class=2 cases (1 of 2 types of calibration) turned out to be easy to identify. Of course, in this setting the operators already know that the calibration event caused the alarm that created the event record, but the idea is still quite useful. It can also sometimes be of interest to better characterize the background also, and in our case there was strong indication of three distinct clusters. We believe that this approach will have multiple applications.

References

1. T. Burr, J. E. Doak, P. Helman, B. Hoffbauer, J. A. Howell, P. M. Kelly, L. McGavran, R. Pitts, J. M. Prommel, T. R. Thomas, R. B. Strittmatter, and R. Whiteson, "Software Toolkit for Analysis Research (STAR)," Los Alamos National Laboratory report LA-12617-MS (1993), pp. 1-102.

2. S. Bleasdale, T. Burr, A. Coulter, J. Doak, B. Hoffbauer, D. Martinez, J. Prommel, C. Scovel, R. Strittmatter, T. Thomas, and A. Zardecki, "Knowledge Fusion: Analysis of Vector-Based Time Series with an Example from the SABRS Project," Los Alamos National Laboratory report LA-12931-MS (April 1995).

3. D. Martinez and T. Burr, "GEXO Data Acquisition Guide," Los Alamos National Laboratory, Safeguards Systems Group report NIS-7/94-1026 (1994).

4. M. Bagshaw and R. A. Johnson, "Sequential Procedures for Detecting Parameter Changes in a Time-Series Model," *J.A.S.A.* **72**, 593-597 (1977).

5. G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, 3rd Ed. (Holden-Day, San Francisco, 1994).

6. C. Chatfield, *The Analysis of Time Series, An Introduction*, 4th Ed. (Chapman and Hall, London, 1989).

7. S. Bleasdale, T. Burr, C. Scovel, and R. Strittmatter, "Knowledge Fusion: An Approach to Time Series Model Selection Followed by Pattern Recognition," Los Alamos National Laboratory report LA-13095-MS (February 1996).

8. T. Burr and R. Strittmatter, "Knowledge Fusion: Comparison of Fuzzy Curve Smoother to Statistically Motivated Curve Smoothers," Los Alamos National Laboratory report LA-13076-MS (February 1996).

9. J. Friedman, "Multivariate Adaptive Regression Splines," *Annals of Statistics* 19(1), 1-141 1991.

10. L. Wang and J. Mendel, "Generating Fuzzy Rules by Learning from Examples," *IEEE Transactions on Systems, Man, and Cybernetics* **22**(6), 1414-1427 (Nov/Dec 1992).

11. T. Burr, "Prediction of Linear and Nonlinear Time Series With an Application in Nuclear Safeguards and Nonproliferation," Los Alamos National Laboratory report LA-12766-MS (April 1994).

12. P. Robinson, "Nonparametric Estimators for Time Series," *Journal of Time Series Analysis* **4**(3), 185-207 (1983).

13. J. Hartigan, *Clustering Algorithms* (Wiley, New York, 1975).

14. J. Banfield and A. Raftery, "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics* **49**(3), 803-822 (September 1993).

15. L. Brieman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees* (Wadsworth & Brooks, Monterey, California, 1984).

16. T. Hastie and R. Tibshirani, "Discriminant Analysis by Gaussian Mixtures," *Journal of the Royal Statistical Society, Series B* (1995).

17. T. Kohonen, "The Self-Organizing Map," *Proceedings of the IEEE* **78**(9), 1464-1480 (September 1990).

18. T. Hastie, R. Tibshirani, and A. Buja, "Flexible Discriminant Analysis by Optimal Scoring," *Journal of the American Statistical Association* **89**(428), 1255-1270 (1994).

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545