MineTool-M^2: An Algorithm for Data Mining of 2D Simulation Data

Tamara B. Sipes and Homa Karimabadi[1,2]
[1] University of California San Diego, La Jolla, CA
[2] SciberQuest, Inc., Del Mar, CA
tsipes@ucsd.edu, homa@eng.ucsd.edu

ABSTRACT
Extraction of knowledge from massive and complex data sets generated from peta-scale simulations poses a major obstacle to scientific progress. We propose a new approach to solving this problem by utilizing an innovative feature extraction technique in combination with specialized data mining algorithms which can be incorporated as part of the scientific visualization pipeline. In this paper we show how data from simulations as well as many other real life examples can be represented in a form of multivariate time series. Accordingly, we have adapted a multivariate time series analysis data mining technique to handle simulation data. The technique extracts global features and metafeatures in the 2D simulation dataset in order to capture the necessary time-lapse information. The features are then used to create a static, intermediate data set that is suitable for analysis using the standard supervised data mining techniques. The viability of the new algorithm called MineTool-M^2 is demonstrated through its application to the problem of automatic detection of flux transfer events (FTE) in the simulation data. MineTool- M^2 built model led to a high FTE classification model accuracy of 95.56% correctly classified instances where the model produced one of three outputs of non-FTE, across cut FTE, and tangent cut FTE. For comparison, two other means of treating the time series data including a common summary statistics technique yielded much lower accuracies of 48% and 62% correctly classified instances, illustrating the complexity of this problem and the need for advanced techniques to handle such data.

Keywords
Multimedia Mining, Temporal and Spatial Data Mining, Multivariate Time Series Classification, Regression/Classification

1. INTRODUCTION
Scientific simulations have been used in a variety of fields to aid the understanding of a variety of scientific processes and enable scientific discovery. In space sciences for example, where progress relies on use of computer simulations in close ties with *in situ* and remote spacecraft measurements, the data challenge is particularly acute and has reached a critical stage. The advent of petascale computing has led to a significant increase in the size of the simulations. Our largest simulations include over 3.2 trillion particles, and 15 billion cells, and are run for several days using 200 K cores on Jaguar. We achieve about 7-9 million particle pushes per core for Cray machines. **Knowledge discovery from these increasingly complex and large data sets is a major bottleneck to progress in a variety of scientific fields today.** There is an eminent need for automated, intelligent methods to enable analysis and knowledge discovery in simulation data.

There are at least a couple of ways one can analyze data utilizing automated approaches and avoiding the time-consuming and error-prone human eye in tracking an event in large simulation data repositories. One obvious way would be to think of a simulation as a series of images, and analyze a 'time series' of image data. This approach would entail an image representation that would encompass the important areas of the image, and presenting it in a series. Another way would be to concentrate on the particular area of the simulation that is of interest and focus on the features being created and changed in time. We adopted the later approach, as it decreases the complexity of the problem, to be able to emphasize the creation and evolution of the events in the simulation, in order to describe them, create a predictive model and obtain the ability to classify them. Our approach entails collecting a certain spatial and temporal information, or features of the event in the simulation window (as in a series of point coordinate values (x,y)), in addition to the other variables available, that describe the (x,y) simulated measurements. These features, or set of points being tracked over time, in effect add another dimension to the time series data at the input. In the sections below we describe how we devise and collect the simulation features as the series of data points, or "cuts" in the example simulation domain, and utilize intelligent data mining classification tools to extract knowledge from them.

In the recent years we introduced a technique called MineTool [10] with distinct advantages over standard data mining techniques. Besides offering high accuracy of the resulting predictive models, a key advantage of MineTool-like approach is that it makes data mining more accessible, by offering a self-contained step by step procedure for model building. MineTool was created to handle static (non-time series) data and further expanded to a multivariate time series analysis technique which is naturally incorporated into the MineTool modeling process, suitable for time series data analysis. Some of the immediate applications of the resulting method, called MineTool-TS (for MineTool-TimeSeries), include multiple event detection and event classification [11].

In this paper, we adapt MineTool-TS to handle simulation data and illustrate its pattern recognition capabilities applied to simulation data. The paper is organized as follows. Section 2 discusses the simulations; Section 3 discusses the time series analysis and the underlying algorithm of MineTool-TS. Section 4 describes the application to simulation data. Summary and discussion are presented in Section 5.

2. SIMULATION DATA
The example that we consider here is the 2D global hybrid simulations (where electrons are modeled as fluid particles, and ions as fully kinetic) of the Earth's magnetosphere [8][9] where interaction of the solar wind plasma and magnetic fields impinging on the Earth's dipole field is modeled. The simulations

are 2D in a sense that only spatial variations of the parameters in two dimensions are retained but all three components of the vectors such as the magnetic field are kept. One feature of particular interest is the so-called flux transfer events [5] which were first observed in spacecraft data and are thought to be magnetic flux ropes formed at the Earth's magnetopause. Many details regarding the FTEs remain poorly understood but peta-scale simulations are enabling us to finally settle many questions regarding their formation, structure, and evolution. Figure 1 shows several examples of FTEs in a 2D global simulation. The simulation box is 2000 x 2000 ion skin depths or about 20 earth radii in each direction. The size of FTEs is small compared to the overall size of the magnetosphere and they appear as regions with density enhancements in this figure. FTEs have complex structures in velocity and magnetic field (not shown). Simulations have one major advantage to spacecraft observations in that one has in effect a very good spatial coverage of FTE at any given time and can track its evolution in time. In contrast, a single spacecraft or even four-spacecraft as in the case of Cluster mission, have limited spatial coverage. Figure 1 shows three sample spacecraft trajectories. Our goal in this particular study was to determine whether data mining algorithms can distinguish between these different cuts which include cuts scheming the surface of the FTE (cut-A), across an FTE (cut-B), or cuts away from FTEs (cut-C). If successful, this would imply that data mining algorithms can equally be applied to spacecraft data to distinguish among these three cuts. It would also imply that there are distinct features among the variables that, for example, would enable the algorithm to distinguish between cuts across and along an FTE.

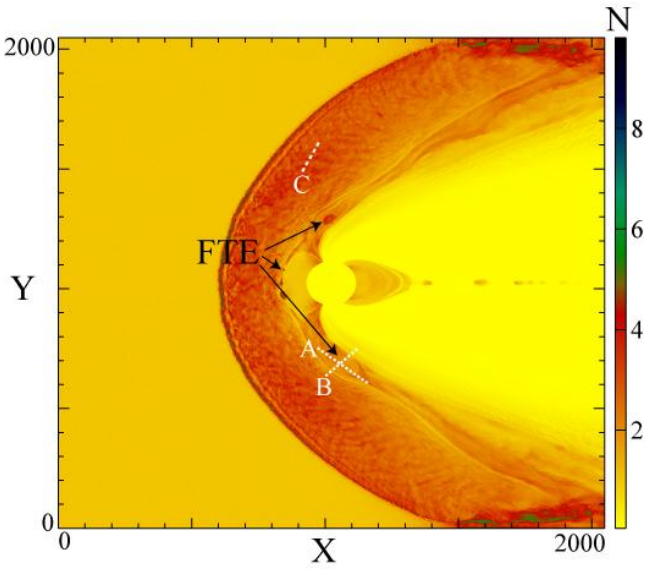

Figure 1. 2D simulation of the Earth's magnetosphere showing three examples of FTEs, and the three sample spacecraft trajectories (A, B and C).

3. TIME SERIES DATA ANALYSIS

3.1 Multivariate Time Series Data

In time series forecasting, one is interested in deciphering and quantifying temporal patterns in the data. In multi-variate time series data analysis, the relationship among the variables, each represented by a time series, can also be important. Time series analysis has become one of the most important branches of mathematical statistics and data mining, and a variety of techniques have been developed. The techniques range from a single time series forecasting (e.g., using the ARIMA method), to time series modification to allow certain patterns to be observed more easily (e.g., using FFT in signal processing), to multivariate time series classification. The latter is the focus of our work presented here.

3.2 Multivariate Time Series Classification

A data mining technique called MineTool-TS was introduced which captures the time-lapse information in multivariate time series data through extraction of global features and metafeatures [11]. In this paper we expand MineTool to handle not only static and time series data, but image, and simulation data as well, and call it MineTool-M^2 for MineTool-MultiMedia.

Time series data containing multiple variables (i.e. multivariate time series data) commonly occurs in a wide variety of fields including biology, finance, science and engineering. A time series (or more generally temporal data) is a sequence of measurements that follow non-random orders and can be generated either from a fixed point measurements at several time intervals or a convolved spatial-temporal variations as measured from a moving detector. Multivariate time series analysis is used when one wants to model and explain the interactions among a group of time series variables such as the field and plasma variables in the space physics domain. Much of the scientific data is in form of multivariate time series. Examples include ECG measurements, *in situ* field and plasma measurements of bow shock crossings, flux transfer events, turbulence in the solar wind, sign language hand movements, among others.

Multivariate time series classification attempts at classification of a new time series based on past observations of time series examples, rather than providing an analysis of a single-variate time series. Just like in any other classification problem, we are given examples of labeled data in order to build a predictive model. Historically, Hidden Markov Models (HMMs), recurrent Artificial Neural Networks (recurrent ANNs) and Dynamic Time Warping (DTW) have been used to build predictive models of multivariate time series data for classification tasks [15][21][24]. Even though these techniques are useful for certain tasks, they have several disadvantages which make them impractical for large datasets. In case of HMMs, for example, the number of parameters that needs to be set and examined is very large, even for small HMMs, determining the number of states for a certain dataset is just an educated guess, leading to many iterations of examining and setting the parameters. HHMs also do not handle continuous values very well, and make several major assumptions not readily available in a real-world scientific dataset. Recurrent ANNs suffer from several of the same problems as HMMs and require the user to experiment and choose many parameters and decide on the appropriate network architecture. The result is also in the form of a black-box which makes it difficult to understand.

If one could replace the time series by a static data consisting of variables that capture the relevant and interesting features (e.g., number of zero crossings, slope, extreme values) of the time series, then the standard MineTool technique could be used. Two

ideas for reduction of time series data immediately come to mind. First, one can randomly select several time instances of the data and treat each instance as a static data. The number of instances selected can be smaller than the total number of time instances available. Second, one can create summary statistics data, i.e., the time series data is replaced by its statistical measures such as the mean, standard deviation, minimum and maximum values, etc. As we will show shortly, even though these techniques are somewhat successful for a small number of simple datasets and problems, neither of these two approaches yields high accuracy results in modeling real life, complex time series data. Instead we use a more sophisticated approach to extract features from multivariate time series data that yields much higher accuracy [7][11].

3.2.1 MineTool for Static Data

The core data mining algorithm that underlies MineTool-TS is MineTool [10]. The advantages of MineTool over traditional algorithms such as support vector machine and artificial neural net (ANN) are its automated steps that make it more accessible and applicable in a variety of domains, accuracy, robustness and the analytical form of the model at the output.

An important algorithmic issue in data mining is how to find the optimal complexity of the model or the fitting function. Too much complexity in the model can result in overfit, whereas not enough complexity can result in underfit. The mathematical foundations of MineTool are based on considerations to balance the competing dangers of underfit and overfit to identify the level of model complexity that guarantees the best out-of-sample prediction performance without ad hoc modifications to the fitting algorithms themselves [14][17][18][26]. MineTool creates a predictive model architecture that is linear in the parameters. The algorithm searches for a model M that best relates rows of the input variable values Xij to the appropriate target value yi (yi = M(Xij)), where i = 1,…,N and j = 1,…,K. The model parameters are either linear combinations of the input ($\mathbf{X_i}$**'α**, where prime indicates transpose of the vector, index *i* refers to the i^{th} observation**)**, linear transformations of the input variables (**ζ*(Xi)***), or highly non-linear transformations of the input (Ψ**(Xi,γ)**). Equation 1 describes the general form of a MineTool model:

$$y_i = \mathbf{X_i'}\boldsymbol{\alpha} + \sum_{p=1}^{P} \zeta(\mathbf{X_i})'\boldsymbol{\delta_p} + \sum_{q=1}^{Q} \boldsymbol{\Psi}(\mathbf{X_i}, \boldsymbol{\gamma_q})'\boldsymbol{\beta_q} \quad (1)$$

In its simplest form, the model would be a linear combination of the input parameters (i.e. a linear regression model). MineTool goes beyond a simple linear model by introducing the linear (such as level-1 and level-2 transformations producing cross-products, ratios, squares, cubes etc.) and non-linear transformation of the input variables, if their addition increases the model accuracy. The non-linear transforms Ψ are single hidden layer feed forward Artificial Neural Net (ANN)-like transforms, just like the ANNs of the same architecture, with the difference that the non-linear transformed inputs are combined into a linear model.

3.2.2 Metafeature and Global Feature Detection

To be able to process a (time) series dataset (represented with multiple rows of data describing one instance or observation) using MineTool, the data needs to be "flattened," or made static. Nevertheless, this needs to be accomplished without losing the important information incorporated in sequential measurements varying with time. Historically, this has been done either by summarizing the data and writing only the mean of the different row values of one observation, or recording the difference between the pairs of rows and then treating them as single instance entries. These techniques work somewhat well on just a limited set of time series problems. For real life, complex scientific datasets, these approaches are most often too weak to incorporate the important time changes in the data. The MineTool-TS solution to this problem is to collect the important time-changing information that can occur in one of the time series variables. While a value varies with time, it most often increases, decreases or stagnates. There are other, more complex features one can record, that consist of the three basic changes, such as bipolar signature (relevant in case of flux transfer events), where a value goes up, then goes down crossing the axis, and goes up again (the sinusoid function has a demonstrates the bipolar behavior, for example). Global features, just like the metafeatures, are used to extract the information from all the rows representing one observation. Global features describe one instance rows using one measurement, such as: the maximum value, minimum values, mean, mode or the number of zero crossings. Some of the metafeatures and global features included in the MineTool-TS algorithm are following:

- **Increasing Metafeature**— An increasing metafeature is recorded for all the consecutive rising time-series measurements. For each increasing event, we record its start point, duration, gradient and average value, so that the increasing events can be used for analysis and comparison.
- **Decreasing Metafeature**— A decreasing metafeature is recorded for all the consecutive reducing time-series measurements. For each decreasing event, similarly to the increasing events, we record starting point, duration, gradient (which is negative in this case) and average value.
- **Plateau Metafeature**— A plateau metafeature is recorded for all the consecutive non-changing time-series measurements. MineTool-TS allows for a small amount of noise to be ignored, so that the true plateaus are captured.
- **Bipolar Signature Metafeature**— A bipolar signature metafeature is recorded for all the consecutive time-series measurements that increase, decrease and cross the zero, and increase again.
- **Global Minimum**—For each single variable, the global minimum feature extracts the minimum value of all of the time observations belonging to one time series instance for the variable, and records it as the global minimum feature for that input channel.
- **Global Maximum**—The maximum value of all of the time observations belonging to one time series instance for the variable, and is recorded as the global maximum feature for that variable.
- **Mean** —The average value of all of the time observations belonging to one time series instance for the variable, and is recorded as the global mean feature for that specific variable.
- **Mode** —The mode value of all of the time observations belonging to one time series instance for the variable, and is recorded as the global mode feature for that specific input variable.
- **Number of Zero Crossings** —Lastly, the number of zero crossings occurring during the time observation recorded measurements is written down as the number of zero crossings global feature.

Next, once all the requested features are collected, the MineTool-TS algorithm performs the feature space segmentation to group

similar features and make them have a higher predictive value for data mining. More details on the algorithm can be found in [11].

3.3 MineTool-M^2 Extension for Multimedia Data Mining

The time series classification algorithm needed to be adapted to handle simulation (and other multimedia) data. Figure 2 illustrates the additions to the basic time series analysis algorithm:

(i) Multimedia (i.e. simulation) data preparation

(ii) Handling of the time series of uneven lengths

3.3.1 Simulation Data Preparation

The simulation data needs to be converted into a series dataset as the algorithm is designed for time series data. A dataset containing a different type of a series data could be entered at the input as well, as the metafeature variables would track the changes that occur either from one point in time to another (as in time series data) or from one point in space to another (as in spatial data). To prepare simulation data for being entered in MineTool-M^2 we perform a preprocessing step (Figure 2) that converts the multimedia data into a series data set. Section 4.2 details our feature extraction step that converts the simulation data into a series "cuts" data and enables further analysis.

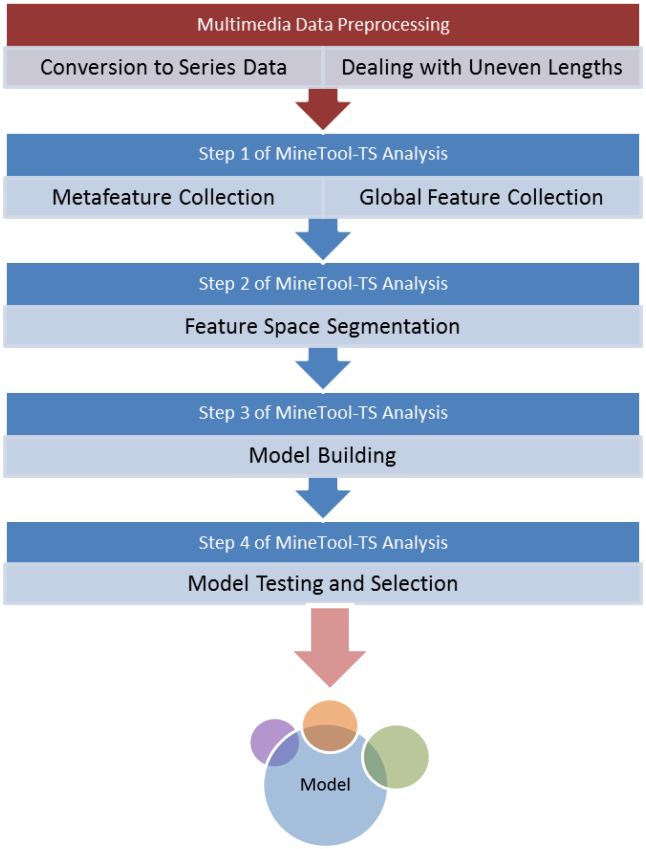

Figure 2. An Illustration of the MineTool-M^2 algorithm.

3.3.2 Time Series of Uneven Lengths

To be able to effectively describe a set of points being tracked in the simulation window, the algorithm needed to be able to handle uneven lengths of the series data. This means that one observation of interest could be tracking an event in the simulation from its creation to, for instance, the time = T_{117}, whereas the other event evolution could end at the time of T_{59}.

Therefore, we adapted the basic method to accept different series observation lengths resulting in MineTool-M^2 expecting an array at the input, listing the number of time observations (NTO) for each of the input series data instances. The original algorithm assumed that all of the series streams are of the equal lengths. Figure 2 illustrates the basic steps of the MineTool-M^2 method. First, the simulation data is converted into a multi-series form by adding a dimension to the expected time series data (resulting in extra columns of the input dataset). Since the data may or may not be of equal length, the algorithm expects an array of the series data lengths at the input. Consecutively, the multivariate time series classification steps of MineTool-TS are performed: metafeature and global feature collection, the feature space segmentation and reduction, following by the iterative model building and evaluation until the best model is selected.

In the following section we illustrate the application of MineTool-M^2 to the Flux Transfer Event (FTE) simulation data.

4. APPLICATION TO SIMULATION DATA

To demonstrate the applicability of MineTool-M^2 to mining time series multimedia data, we looked at the problem of automatic detection of Flux Transfer Events (FTE) in simulation data.

FTEs are typically identified on the basis of clear isolated bipolar signatures in the B_n component of the magnetic field (in the LMN coordinate system). The Cluster spacecraft magnetic field observations of 4-s resolution from the Fluxgate Magnetometer (FGM) [1] and plasma observations of 4-s resolution from the Cluster Ion Spectrometry (CIS) instrument [22] are commonly used for Cluster magnetopause crossings and FTE identifications. The measurements include a total of 11 input variables: B_x, B_y, B_z, $|B|$, N_p, V_x, V_y, V_z, $T_{\|}$, $T_{\perp}$,T_t However, simulation is used to enable visualization of what the collected measurements mean, how these events occur in magnetosphere, and aid the scientist in evaluating novel algorithms and gaining better understanding of these events.

4.1 Description of the Test Problem

FTEs are usually detected based on the data signatures tangent to the FTE events. The goal of the data analysis and modeling is to build a model that will be able to distinguish the cuts across the FTEs from the cuts tangent to FTEs (two classes), as well as differentiate non FTEs. This is a challenging three-class, multivariate data series classification problem. In FTE observations, scientists can identify FTEs only by looking at signatures tangent to FTEs and our goal is to, using simulation and the presented MineTool-M^2 approach to data mining of multimedia time series data, improve this fact.

4.2 Data Collection and Preparation: "Cuts" Feature Extraction

To analyze simulation data in tracking an event, we concentrate on the particular area of the simulations that is of interest and focus of these features being created and changing in time. In this manner, we are able to emphasize the FTE events in order to describe them, model and classify them. We introduce the *"cut" feature*, a novel computer vision feature extraction method that enables us to collect the important characteristic of the area of interest within simulation data window, while decreasing the complexity of the data selected for further analysis. A "cut" or a "slice feature" is a line drawn at the site of the feature of interest, or at the site of the feature non-existence. "Cuts" are modeled based on the *spacecraft trajectories* and, in effect, simulate what a spacecraft would observe while on a trajectory near an event or non-event. Our goal here was to determine whether data mining algorithms can distinguish between these different cuts. We have devised a cutting routine for making "cuts" or "slices" in the simulation data and creating a data file to be used in analysis and modeling. Figures 3a, 3b and 3c show three sample spacecraft trajectories-guided cuts or slices in the 2D simulation data which include cuts scheming the surface of the FTE (cut-A), across an FTE (cut-B), or cuts away from FTEs (cut-C). The variables that were observed in the cuts included:

X, Y, B_X_slice, B_Y_slice, B_Z_slice, Density_slice, T_{PAR}_slice, T_{PERP}_slice, T_{TOTAL}_slice, V_{IX}_slice, V_{IY}_slice, V_{IZ}_slice, B_{TOTAL}, event

The simulation FTE data has been labeled with three labels: a) cuts tangent the FTE, b) cuts across to the FTE, and c) non-events. The dataset consists of series data and does not have to have the same length. In this phase of the project, we collected 45 of each of the types of FTE events, giving 135 total events, or streams of data. Each of our events had up to 1000 data points representing one cut, however the length was varying. We have prepared the data and converted in the form suitable for mining using our MineTool-M^2 method for multivariate classification of multimedia time series data.

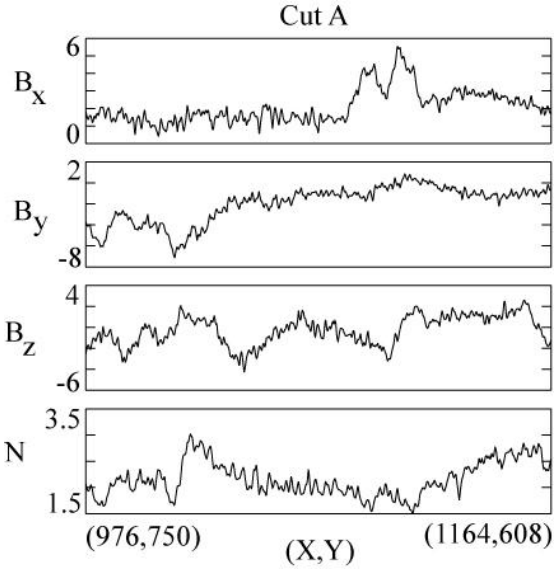

Figure 3a. A Cut in the Simulation Data Tangent to the FTE.

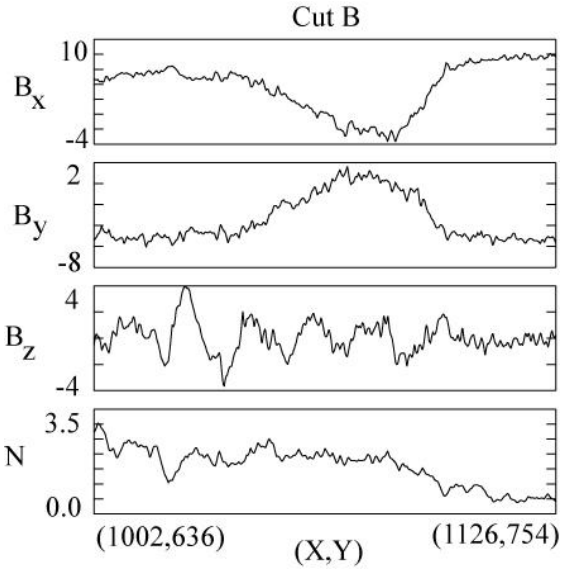

Figure 3b. A Cut in the Simulation Data Across the FTE.

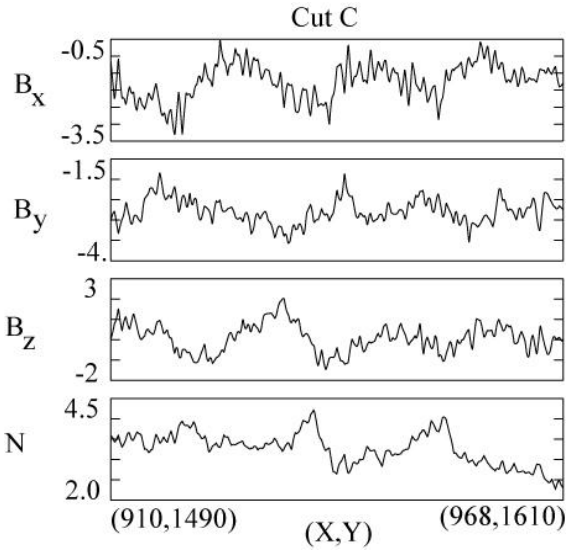

Figure 3a. A Cut in the Simulation Data Away From the FTE.

4.3 Modeling Results

Our method first converted the simulation cuts data into series data, followed by going through the series data and collecting the metafeature information, such as increases, decreases and plateaus in one series. Then, using this information each of the series was "flattened" into a static row of data and fed into the intermediate dataset. This was completed for each of the 135 event examples. The flattened, static dataset was then fed into our MineTool algorithm, to discover the correlations among the input variables to the output variables.

We contrast the modeling results of the flux transfer event (FTE) classification in simulation data performed in three different ways (as listed in Table 1): as a static dataset (where each row is treated

as an independent instance, and not as a part of a series), as a series data, using the summary statistics representation, where a series is converted into a single instance of data using measurements such as mean, standard deviation, minimum, maximum, range, number of zero crossings, interquartile range (or, the spread) and the median value, for each of the variables in the data), and as the true series data, using MineTool-M^2. The MineTool- M^2 approach requires the most computational time, followed by the summary statistics approach and static analysis approach. However, the accuracy of the model (96.6% vs. 62.4% vs 47.6%) is a worth-while trade off. Table 1 describes the results obtained in our study using standard data mining evaluation statistics (percentage of correctly classified instances, correlation coefficient, mean absolute error (MAE) and root mean squared error (RMSE)).

```
      event = 0.941734
      + 39.9488*Btotal_Inc_14*den_Inc_5
      -3.51784*vix_Dec_5
      -359.589*tperp_Dec_4*tperp_Dec_6
      -57.0447*tperp_Inc_2*viy_Dec_5
      -5.70272*tperp_Inc_3
      + 392.001*ttotal_Inc_10*ttotal_Dec_25
      + 53.5713*ttotal_Inc_20*viy_Inc_2
      -14.6957*tperp_Inc_2*tpar_Dec_4
      + 102.018*Btotal_Inc_10*vix_Dec_3
      + 95.3943*den_Inc_8*bx_Dec_22
      -61.8677*viz_Inc_9*ttotal_Dec_7
      -15.0103*Btotal_Inc_5*den_Dec_16

Where :
Btotal_Inc_14 represents a time series
feature with the following average
description:
      average value of -> 0.330258
      mid time value of -> 552.96
      gradient value of -> 0.002956
      duration of -> 64.3565

den_Inc_5 represents a time series feature
with the following average description:
      average value of -> 0.258179
      mid time value of -> 543.827
      gradient value of -> 0.00134285
      duration of -> 25.4539

vix_Dec_5 represents a time series feature
with the following average description:
      average value of -> 0.347282
      mid time value of -> 747.303
      gradient value of -> -0.00113456
      duration of -> 40.228  etc.
```

Figure 4. The Predictive Model of FTEs.

The modeling results are producing a model with 95.6% accuracy tested on a third of the data, set aside as holdout (test) data, and built on the 66% of the data as the training set, with each of the classes being equally represented in the training and test data. The model picks up on the most important metafeatures in the classification of an event as an across FTE, tangent FTE or non-event, and is given in Figure 4.

The predictive model created by the MineTool data mining method is in an analytical form, enabling insight into the most important metafeatures and global features detected by the algorithm in appropriately classifying a time series instance of data. The model in Figure 4 shows that the specific total magnetic field (B_{TOTAL}) together with the specific increments in density (which is a level-1 cross product linear transformation $\zeta(Xi)$) from the Eq. 1) positively correlates to a series cut variable being classified as an FTE, while if the V_X_Dec_5 (a simple linear combination of the input variable X_i'α) is detected, it negatively correlates with an FTE event (there were no highly non-linear transformations $\Psi(Xi,\gamma)$ in the model chosen by the method). The model is also able to very accurately distinguish between an event label 1 and 2 (across and tangent FTE).

Table1. Comparative analysis of MineTool-M^2 vs. other methods.

Type of Analysis	Correctly Classified	Correlation coefficient	MAE	RMSE
Static Data Analysis	47.6%	0.376235	0.5682	0.6932
Summary Statistics Analysis	62.4%	0.554823	0.4951	0.6351
MineTool- M^2	95.6%	0.952676	0.1772	0.2532

5. SUMMARY AND DISCUSSION

In this paper we aim to contribute to the urgent need to understand and learn from the often massive, constantly increasing, complex, multimedia data, often collected or created in the form of simulation data, in an automated fashion.

We adapt our multivariate time series analysis data mining technique to handle simulation data. We extract the important information from the simulation data by introducing a novel computer vision feature extraction operator named "cuts" that collect the cuts-type of data in the simulation window. The cuts-data are then converted into a series data and input into MineTool-M^2 for analysis and modeling. We also expand the method to allow for uneven lengths of the series data at the input. The technique extracts global features and metafeatures in the 2D simulation dataset in order to capture the necessary time-lapse information. The features are then used to create a static, intermediate data set that is suitable for analysis using the standard supervised data mining techniques.

The capability of the new algorithm called MineTool-M^2 is demonstrated through its application to the problem of automatic detection of flux transfer events (FTE) in the simulation data. MineTool- M^2 built model led to a high FTE classification model accuracy of 95.56% correctly classified instances where the model produced one of three outputs of across cut FTE, tangent

cut FTE, and non-FTE. For comparison, two other means of treating the series data including a common summary statistics technique yielded much lower accuracies of 48% and 62% correctly classified instances, illustrating the imminent need for advanced techniques, such as MineTool-M^2, to handle such data.

Our future work will encompass the expansion of MineTool-M^2 to 3D simulation data, and other multimedia data as well. By applying and extending ideas from data mining, image and video processing, statistics, and pattern recognition, we are developing a new generation of computational tools and techniques that are being used to improve the way in which scientists extract useful information from data.

6. ACKNOWLEDGMENTS

This work was supported by a NASA SBIR and NNX11AC83 grant at SciberQuest, Inc., Simulations were performed on Kraken, a Cray XT5 system provided by the National Science Foundation at the National Institute for Computational Sciences, and on NASA' s Pleiades, which is provided by the NASA High-End Computing (HEC) Program.

7. REFERENCES

[1] Balogh A, Dunlop MW, Cowley SWH, Southwood DJ, Thomlinson JG, Glassmeier KH, Musmann G, Luhr H, Buchert S, Acuna MH, Fairfield DH, Slavin JA, Riedler W, Schwingenschuh K, Kivelson MG, The Cluster magnetic field investigation, *Space Sci. Rev*., 79, 65-91, 1997.

[2] Candes, E.. *Ridgelets: Theory and Applications*. PhD thesis, Stanford University, Department of Statistics, 1998.

[3] Cortes C. and Vapnik V. Support-Vector Networks, *Machine Learning, 20, 1995.*

[4] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1-38.

[5] R. C. Elphic. Observations of Flux Transfer Events: A Review. the American Geophysical Union, 1995.

[6] Hartley, H. (1958). Maximum likelihood estimation from incomplete data. *Biometrics*, 14:174–194.

[7] Kadous, M. W. *Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series*. PhD thesis, School of Computer Science & Engineering, University of New South Wales, 2002.

[8] Karimabadi, H., and J. Dorelli, H. X. Vu, B. Loring, Y. Omelchenko, Is quadrupole structure of out-of-plane magnetic field evidence of Hall reconnection?, to appear in *Modern Challenges in Nonlinear Plasma Physics*, editor D. Vassiliadis, AIP conference, 2010.

[9] Karimabadi, H., H. X. Vu, D. Krauss-Varban, Y. Omelchenko, Global Hybrid Simulations of the Earth's Magnetosphere, *Numerical Modeling of Space Plasma Flows: Astronum-2006*, vol. 359, 257, 2006.

[10] Karimabadi, H., Sipes, T. B., White, H., Marinucci, M., Dmitriev, A., Chao, L.K., Driscoll, J., Balac, N. (2007). Data Mining in Space Physics: 1. The MineTool Algorithm, *J. Geophys. Res*., 112, A11215, doi:10.1029/2006JA012136.

[11] Karimabadi, H., Sipes, T. B., Wang, Y., Lavraud, B. and Roberts, A. (2009). A new multivariate time series data analysis technique: Automated detection of flux transfer events using Cluster data, *J. Geophys. Res*., *Vol 114, A06216, doi:10.1029/2009JA014202, 2009*

[12] Lendasse, A., Lee, J. , de Bodt, E., Wertz, V. and Verleysen, M.. Approximation by Radial Basis Function Networks Application to Option Pricing. In Connectionist Approaches in Economics and Management Sciences, C. Lesage, M. Cottrell eds., Kluwer academic publish., 2003, pp. 203-214

[13] Looney, C. G., Pattern recognition using neural networks, Theory and algorithms for engineers and scientists, Oxford University Press, 1997.

[14] Marinucci, M., Automatic Prediction and Model Selection,Ph.D. Thesis, Departamento de Fundamentos del Analisis EconomicoII, Facultad de Ciencias Economicas, Universidad Complutense de Madrid 2007.

[15] Myers, C. S. and Rabiner, L. R. A comparative study of several dynamic time-warping algorithms for connected word recognition. The Bell System Technical Journal, 60(7):1389-1409, September 1981.

[16] McLachlan, G. and Krishnan, T. (1997). *The EM algorithm and extensions*. Wiley series in probability and statistics. John Wiley & Sons.

[17] Pérez-Amaral, T., Gallo, G. M. and White, H., A Flexible Tool for Model Building: the Relevant Transformation of the Inputs Network Approach (RETINA), *Oxford Bulletin of Economics and Statistics*, 65 (s1), 821-838, 2003.

[18] Pérez-Amaral, T., Gallo, G. M. and White, H., A Comparison of Complementary Automatic Modeling Methods: RETINA and PcGets," *Econometric Theory*, 2005.

[19] Powell, M. J. D. *Radial basis functions for multivariate interpolation: A review*. In Algorithms for Approximation, J. C. Mason and M. G. Cox, Eds. Clarendon Press, Oxford, 1987.

[20] Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

[21] Rabiner, L. R. and Juang, B. H. An introduction to hidden markov models. IEEE Magazine on Acoustics, Speech and Signal Processing, 3(1):4-16, 1986.

[22] Reme, H. and C. Aoustin and J. M. Bosqued and I. Dandouras, B. Lavraud, et al., First multispacecraft ion measurements in and near the Earth's magnetosphere with the identical Cluster ion spectrometry (CIS) experiment, *Annales Geophysicae*, 19, 1303-1354, 2001.

[23] Ripley, B.D., Pattern Recognition and Neural Networks, Cambridge University Press; 1996.

[24] Schmidhuber, J., Graves, A., Gomez, F. and Hochreiter, S. Recurrent Neural Networks, Cambridge University Press, 2012.

[25] White, H., Approximate nonlinear forecasting methods, in *Handbook of Economic Forecasting*, Volume 1, Edited by Elliott, Granger and Timmermann, Elsevier, Amsterdam, 2006.

[26] White, H., Personnel Readiness: Neural Network Modeling of Performance-Based Estimates, *Final Report to the Office of Naval Research, Contract #: N00014-95-C-1078*, 1999.

[27] R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. *21(3).* pp. 876—880. Available: http://www.halcyon.com/pub/journals/ 21ps03-vidma