©Copyright 2018

Jewel YunHsuan Lee

Machine Learning of Spatiotemporal Bursting Behavior in Developing Neural Networks

Jewel YunHsuan Lee

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Computer Science and Software Engineering

University of Washington

2018

Reading Committee:

Michael Stiber, Chair

Erika Parsons

Dong Si

Program Authorized to Offer Degree:
Computing and Software Systems

University of Washington

Abstract

Machine Learning of Spatiotemporal Bursting Behavior
in Developing Neural Networks

Jewel YunHsuan Lee

Chair of the Supervisory Committee:
Professor Michael Stiber
Computing and Software Systems

Experimental investigation of the collective dynamics in large networks of neurons is a fundamental step towards understanding the mechanisms behind signal and information processing in the brain. In the last decade, the emergence of high performance computing technology has allowed long-duration numerical simulations to model large-scale neural networks. These simulated networks exhibit behaviors (ranging from stochastic spiking to synchronized bursting) that are observed in living preparations. These simulations' high spatiotemporal resolution and long duration produce data that, in terms of both quantity and complexity, challenge our interpretative abilities. This thesis presents an application of machine learning techniques to bridge the gap between microscopic and macroscopic behaviors and identify the small-scale activity that leads to large-scale behavior, reducing data complexity to a level that can be amenable to further analysis.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

I would like to express my deepest gratitude and appreciation to my respected advisor, Professor Michael Stiber, for the enlightening guidance and constant support throughout my study and research. This thesis would not have been possible without his incredible patience, immense knowledge, and insightful feedback.

I thank my committee members, Professor Dong Si and Professor Erika Parsons, for their precious time and inspiring comments. My grateful thanks to the members of UWB Biocomputing Lab for their helpful discussions and to Sinchai Delong and Derek Ogle for their timely technical support. I would also like to acknowledge the UWB Gray Hats for sharing computational resources and the University of Washington Bothell School of STEM/CSS Division for providing financial support through the Graduate Student Research Assistantship.

I am very thankful to my family and friends for their incessant encouragement and lovely company. A special thanks goes to Jonas McCoy, who has spent his own time to proofread and improve upon my writing.

Last of all, I am forever grateful to my supportive husband, Alan Lee.

DEDICATION

To my beloved mother, Lucia Chen,
and my amazing husband, Alan Lee,
for being my pillars of support.

Chapter 1
BIOLOGICAL BACKGROUND

One of the central goals of neuroscience is to understand how the brain stores and transmits information. Numerous electrophysiological studies have led to recognize that nervous systems convey information through neural activity patterns; however, it has not resolved into a clear picture of how functional neural networks operate as information processing systems. Although this is a long-term task, piecemeal efforts in deciphering neural activities in the brain still lead to progress in many areas, such as understanding neural pathologies, designing neural prosthetics, and inspiring artificial intelligence models. The following sections in this chapter provide a very brief introduction to several elementary notions of neurosciences and important dynamical properties of neural networks. The aim of this chapter is to provide the reader with minimum information necessary to relate the topics covered by this thesis.

1.1 Neuroscience Basics

1.1.1 The Nervous System

The *nervous system* is a complex network composed of specialized cells that carry messages between the brain and the body. In most types of organisms, the nervous system can be divided into two major parts — the *central nervous system (CNS)* and the *peripheral nervous system (PNS)*; the former includes the *brain* and *spinal cord* whose main function is to integrate and process sensory information, and the latter includes a large system of *nerves* which are long fibers that connect the CNS to every other part of the body.

As an essential communication system inside the human body, the nervous system performs three main functions: gathering sensory input, integrating that input internally, and communicating proper motor output; these functions are completed by a number of specialized cells, or *neurons.* Based on their roles, neurons found in the human nervous system can be divided into three classes: *sensory neurons* that convert external stimuli from the surroundings and send signals to the CNS, *motor neurons* that receive signals from the CNS and covey commands to muscles, organs and glands, and *interneurons* that connect one neuron to another within the CNS.

1.1.2 Neurons and Synapses

Fundamentally, *neurons*, or *nerve cells*, are the basic functional units of the nervous system that receive and transmit information by electrochemical signaling. A typical neuron consists of three functionally distinct parts: the cell body, *soma*, and the processes that extend outward from the cell body, *dendrites* and *axon.* A dendrite is where a neuron receives signals from other cells and transmit them to the soma. The soma integrates input signals and determines if an output signal should be generated. If the output signal is generated, it travels along the axon toward other neurons. Figure 1.1 shows the structure of a neuron.

In neuroscience, the sending neuron and the receiving neuron are commonly known as *presynaptic* neuron and *postsynaptic* neuron, respectively. The connecting point of where the axon of presynaptic neuron communicate with the dendrite of the postsynaptic neuron is called a *synapse*, an electrical or electrochemical signal junction between neurons. As far as we know, there are more than 100 billion neurons in the human brain and each of them can have more than 10,000 synaptic connections with other neurons through synapses.

1.1.3 Neuronal Spikes

The signals that are used in communication among neurons are sequences of electrical pulses that are known as *action potentials*, or *spikes* (Figure 1.1). The generation of spikes depends on a neuron's *membrane potential*, an electrical potential (voltage) difference between the inside of the neuron and its surroundings. Typically the membrane potential rests at about $-70mV$ but can be altered by incoming signals. Some signals are *excitatory* inputs that cause positive change in the membrane potential and make a neuron more likely to

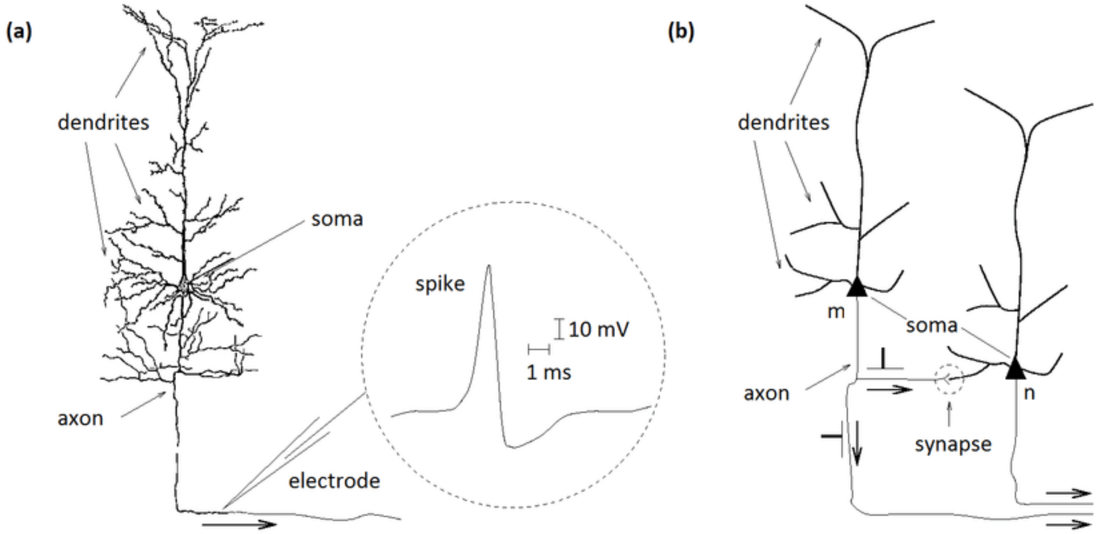


Figure 1.1: (a) Single neuron in a drawing by Ramon y Cajal [2]. Dendrite, soma, and axon can be clearly distinguished. The inset shows an example of the neuronal action potential observed on an electrode. The action potential (spike) is a short voltage pulse of 1-2 ms duration and an amplitude of about 100 mV. (b) Signal transmission from a presynaptic neuron m to a postsynaptic neuron n through the synapse (dashed circle). The axons at the lower right end lead to other neurons.

generate a spike; as opposed to *inhibitory* inputs which cause negative change and make a neuron less prone to spike. Only when the summation of input signals reaches above the threshold value, the neuron sends out an action potential, or fires a spike. After a spike is fired, there is a period of time when the membrane potential becomes transiently more negative than the resting potential and the neuron is incapable of generating another spike until the membrane potential recovers to its normal resting potential. This is known as the *refractory period* of a neuron.

A spike does not vary in amplitude like a graded potential. Instead, it obeys the *all-or-none law* where a neuron either does not reach the firing threshold or a full spike is generated (Figure 1.2). Since the shape (voltage waveform) and size (magnitude) of spikes are stereotypical, they are assumed to carry minimal information; therefore, it is considered that information is encoded in the spike rate or in the spike timing. Arguably neuronal spikes are viewed as binary events, in which 0 represents no spike and 1 represents a spike for each time step. An analogy of information processing in neuron networks is therefore commonly used — neurons communicate with spikes as computers manipulate with bits.

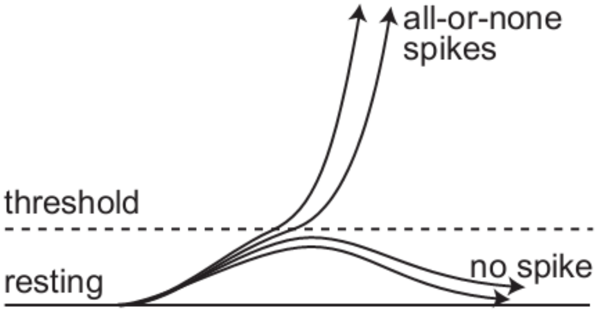


Figure 1.2: Schematic representation of the all-or-none spikes [1]

1.2 Neuronal Network Behaviors

Over time, experimental studies have given us a variety of observations on cortical neural networks. A number of behavioral features of network dynamics have been identified, including whole network bursting, spike wave propagation, and neuronal avalanches.

1.2.1 Whole Network Bursting

The most striking behavior of network dynamics in cultured cortical networks is network bursting. A *network burst* or *population burst* is defined as a synchronized spiking event that involves most or all of the neurons throughout the network. Previous investigations showed that network bursts are a major component of cultures' activity patterns in developing networks of cortical neurons. Wagenaar et al. [3] found that after two weeks *in vitro*, the network activity of cortical cells was dominated by network bursts, but the sizes, shapes, and temporal patterns of these bursts changed widely at different stages of the maturation process. Other studies also agreed that network bursts began to occur and varied as the development progressed, repeating at irregular intervals and developing increased intensity and decreased duration [4, 5, 6].

1.2.2 Neuronal Avalanches

This whole-network bursting is but one aspect of emergent behaviors observed in living preparations and simulations. Another feature that has been noted in spiking activities is a power-law relationship in the propagation of network activity that shares characteristics with systems being in a state of *self-organized criticality (SOC)*. SOC describes that large interactive systems naturally evolve towards a critical state in which any perturbation is capable of triggering cascades of events (or avalanches) throughout the system, and these events can be well-described by power laws. It is a common phenomenon observed in certain complex systems with multiple interacting components, for example, neural networks, forest fires, and earthquakes [7, 65].

Experiments on cortical cultures and slices have produced results consistent with power-law distributed avalanche sizes[8, 9, 10]. The term *neuronal avalanches* is therefore proposed to describe are runs of sequential spikes that occur more frequently than the overall average and are shown to exhibit power-law distributions of avalanche sizes probability. In addition, simulations of small networks have demonstrated development of neuronal avalanches with similar power-law probability distributions [11, 12, 10].

1.2.3 Spatiotemporal Spike Wave Propagation

To investigate spatiotemporal patterns of spiking activity, one important question is how neural activity propagates through cortical networks that are connected through synapses. There has been documentation about propagating waves of neural activity observed via multi-electrode recordings in a number of different neocortical areas including visual [13, 14], auditory [15] and motor cortex [16]. In recent work, spike propagation was observed as a cluster of excitation waves in both simulated and cultured neuronal networks [17]. This phenomenon is called *spike wave propagation*, and describes how the neural activity patterns are organized spatiotemporally as synchronous waves which propagate from one site to neighboring sites in all directions. In addition, spike propagation results showed that the propagation route and speed changed in response to alterations in the variances of the synaptic propagation delay and refractory period where larger variances resulted in faster propagation of spikes [18].

Chapter 2
NEURAL DATA ANALYSIS BACKGROUND

There are multiple factors which simultaneously affect the activity of neurons and collectively influence the overall network dynamics. In modern neuroscience, neurons are recognized as information processing units that perform complex computations on inputs including, but not limited to, past history, reflecting biophysical properties (such as refractoriness), and the activities of neighboring neurons that are synaptically connected. Determining such computation is extremely difficult when we often know the output (the spikes) but have little information on the input (synaptic activities). Hence, scientists must draw inferences about the computation from its results, namely the output spike train data. In this chapter, we present methods and challenges in spike train data analysis from literature review and define the problem and goal for this thesis.

2.1 Spike Train Data

Spiking activity that no longer contains amplitude information and is recoded solely in the sequence of times is called a *spike train*. A spike train is a realization of a point process where a series of point events occur in time, separated by random intervals. An ensemble of spike trains from simultaneously recorded neurons is a spatiotemporal point process where each data point represents the time and location of a spike event.

To understand spike train data, both at the level of individual neurons and in groups of neurons, one approach is to study cultures of cortical cells grown on multi-electrode arrays (MEAs) [19, 20, 21]. These cultures allow recordings of individual neurons and local groups of neurons during development. MEAs can capture the time-varying extracellular potential at each recording location; spikes are then identified in the recording and discriminated to specific neurons based on the shape of the action potential wave-form. These two steps are called *spike detection* and *spike sorting*, respectively. The spike train data collected from simultaneously recorded neurons promises to yield insight into how neurons interact and influence one another.

Another approach is to model neuronal networks and record spike train data from numerical simulations [22]. Such simulations exhibit many of the same behaviors that are consistent with those observed in living preparations. Gritsum, le Feber, and Rutten [6] developed a detailed model that integrated growth-activity model and network topology; their simulation results yielded realistic bursting patterns. Simulations also allow scientist to investigate how different system parameters affect the output activities, potentially result in a better understanding of network behaviors or better designs for subsequent simulations.

2.2 Traditional Analysis

In traditional spike train data analysis, a variety of point process statistical methods are applied to electrophysiological recordings for identifying statistically important temporal or spatiotemporal patterns and their associations [23]. The *auto-correlation* function describes the probability of emitting a spike as a function of the time elapsed from the preceding spike in the same spike train [24]. The *cross-correlation* of two simultaneously recorded spike trains can be quantified using the *cross-correlogram*, which describes the probability of observing a spike in one spike train as a function of time before or after a spike in some other spike trains [25]. The *cross-intensity* function estimates the spike rate of one neuron at different lags relative to the spiking activity of a second neuron [26, 27]. These methods only measure the paired associations in neural activity. To evaluate higher-order neural interactions, greater than pairwise, for example, *spike coincidences* assess the statistical significance of the occurrences of similar spiking patterns among two or more neurons to identify groups of patterns that are worthy of further analysis [28, 29]. *Maximum likelihood* and *maximum entropy* methods have also been popular in predicting higher-order network correlations [30, 31, 32, 33, 34].

These traditional statistical methods have worked well when applied to small numbers of neurons and restricted time frames to identify statistically conspicuous spiking patterns. However, since the ability for neuronal networks to process information derives not only from neurons' individual abilities to generate temporal sequences of spikes, but also from their collective dynamics at the network level [35], it is essential to study activities from a larger number of neurons (thousands or more). With the number of observed neurons increases, the number of possible interactions, or patterns, grows rapidly as well. Traditional statistical methods become unwieldy when they aim to count the occurrences of each of these patterns and distinguish them from chance patterns, demonstrating that the total number of possible patterns is simply too large for an exhaustive search. For instance, pair-wise interactions between 100 neurons requires 10,000 parameters. To precisely identify these patterns from recordings hours to days in length, during which spikes happen in the millisecond time scale, is not computationally feasible. The need for algorithms and analysis tools that scale well with the growing data size and complexity has become an emergent issue.

2.3 Machine Learning Analysis

Recent progress in electrophysiological recording technology has enabled the number of parallel spike trains to grow rapidly (currently allowing a hundred or more neurons to be simultaneously recorded), doubling every 7 years, mimicking *Moore's law* [36]). Similarly, the utilization of *general purpose graphics processing unit (GPGPU)* has allowed neural simulations to model tens of thousands of neurons in extremely long-duration simulations and complete in reasonable time periods (days or weeks, instead of months or years) [37, 38, 39]. These development of high-throughput data acquisition technologies and high performance simulations has begun to transform neuroscience and other modern science (and their computational counterparts) from a data-poor into a data-rich science. In addition, the availability of rich data sets has led researchers from classic hypothesis-driven approaches to innovative data-driven approaches when they can no longer comprehend the complexity and high dimensionality of data by human perception alone.

In dealing with complex, large, high dimensional data, machine learning (ML) has steadily become more successful in extracting key features and relationships which might be difficult to be discovered using traditional statistics. ML utilizes a variety of statistical, probabilistic and optimization tools to learn and improve performance from new data and past experiences without being explicitly programmed. Over the past 20 years, ML techniques such as *regression*, *Bayesian statistics*, *regularization* and *dimensionality reduction* (or *feature reduction*) have been applied extensively in the field of bioinformatics [40, 41]. In neural data analysis, *principal component analysis (PCA)* and *linear discriminant analysis (LDA)* have been popular for linear dimensionality reduction, but further analysis is still needed for assessing the meanings and representations of components or combination of features [42, 43, 44, 45]. In event relationship analysis, *formal concept analysis (FCA)* and *frequent item-set mining (FIM)* are commonly used in extracting frequent patterns of spike activities; however, finding the cause for these patterns has continued to pose a problem [46, 47, 48].

While these unsupervised methods contribute to learn alternative, low-dimensional representations of the neural data by labeling or grouping reoccurring spatiotemporal events for pattern recognition, supervised learning methods have gradually gained popularity in *neural encoding and decoding* to characterize the relationship between the stimulus and activity of individual neuron or neurons in the ensemble [49]. Neural encoding is to map from stimulus to response and construct models that attempt to predict responses to other stimuli; neural decoding is to reverse map and obtain a reconstruction of the original stimulus. In this domain, ML offers predictive models such as support vector machine (SVM), decision tree (DT), artificial neural network (ANN) that are able to map non-linear heterogeneous input and output patterns when the physiological relationships are unclear [50, 51].

On the other hand, one area of neural data analysis in which ML has become indispensable is in spike detection and spike sorting. ML classification and regression methods such as Support Vector Machine (SVM), K nearest-neighbor (KNN) or K-means clustering were used to classify spikes by their shapes (wavelets) and automate the process, improve the accuracy of spike identification and classification (which critically affect the subsequent spike analysis) for the increasingly complex data from neural recordings [52, 53, 54, 55]. One important message we got from these previous work was that while there are a broad range of ML methods with different advantages for neural data anlayisis, the decision about which one is appropriate depends on the requirements of the task.

2.4 Purpose of the Work

The goal of this thesis is to develop simple and meaningful spatiotemporal representations from raw spike train data and apply a set of machine learning techniques to locate burst trigger pattens from large amount of spike events. Our ideas are constructed from the following assumptions and hypotheses:

1. Large spatiotemporal neural dataset are an important consideration for emerging data analysis techniques not only with the quantity but also the complexity of the data.

2. General machine learning methods can, in a relatively straightforward manner, use simple representations of the neural data to investigate major dynamical behaviors in the network.

3. In the large high-dimensional data, only a subset of the neural dimensions are relevant for triggering major network events. These relevant dimensions need to be identified and combined in a manner in which information about the event is maximized.

With these hypotheses, the proposed approaches serve as a mean to reduce data quantity and complexity, improve data visualization, and connect macroscopic behaviors to the relevant small-scale activities of individual cells, aiding researchers in their quest to understand the neural data from the increasingly high-dimensional datasets. In particular, we are interested in applying machine learning techniques to see if stereotypically brief, localized patterns of activity that trigger network bursts can be identified among many millions of spikes from many thousands of separate spike trains.

Chapter 3

DATA ACQUISITION METHODS

Previous studies have shown that it is crucial to investigate how groups of neurons interact with each other, and how their individual spiking activities evolve into generating profound network behaviors. Since there are obvious physiological barriers that keep us from gathering detailed information from collective dynamics of large networks of neurons, understanding such activities is hardly possible without numerical studies.

We wanted to collect simulated spike data with millisecond-scale resolution from large neural population (at least 10^4 cells) over long periods of development (days to weeks) to approximate the size and time scale in living preparations. At the same time, it is additionally necessary to strike a balance between using simple computation models to decrease network complexity and maintaining the essential network features. Kawasaki and Stiber [56] proposed a simple model of cortical culture growth which includes the minimal dynamics at each level necessary to reproduce network behaviors that are seen in the living systems. Most importantly, they developed a GPU-enabled neural simulator for closed-loop, MEA-scale simulations to model the entire network development at the temporal resolution of individual neuron spiking activities with the option of recording the location of each spike to provide spatial resolution. In this thesis work, we reproduced simulations from [56] as our spatiotemporal dataset. In this chapter, a brief introduction to the models used in [56] was given to reiterate the basic properties of neuron and spike generation that were previously discussed in chapter 1; simulation configurations and details are also explained.

3.1 Computation Model

In [56], they used a *leaky integrate-and-fire model* that includes synaptic, constant bias, and noise currents [57]. According to *Kirchoff's law*[1], the model is formalized as

$$C_m \frac{dV_m}{dt} = \frac{1}{R_m}(V_{\text{rest}} - V_m) + I_{\text{syn}} + I_{\text{inj}} + I_{\text{noise}} \tag{3.1}$$

with V_m and V_{rest} being the membrane potential and resting membrane potential, C_m and R_m being the membrane capacitance and resistance, and I_{syn}, I_{inj}, I_{noise} representing the total synaptic current, constant depolarizing current and noise current, respectively. When V_{m} exceeds the firing threshold, V_{thresh}, a spike is generated. V_{m} is then reset to V_{rest} and maintain the value for T_{refract} duration to approximate the *absolute refractory period.* Generated spikes are then transmitted to connected neurons via synapses. The synapse and neurite outgrowth models that determine the network connectivities were explained in detail in [56].

3.2 Simulation Configuration

The BrainGrid neural simulator [39] was used in this work to produce spike data for a network of 10,000 neurons simulated and recorded for 600 million temp steps with 0.1 ms resolution, that is equivalent to 28 days *in vitro.* The network is arranged in a 100×100 rectangular grid with inhibitory, excitatory, and endogenously active neurons uniformly distributed; the neuron distribution layout refer to [56]. Simulations in [56] used this network layout and varied the growth parameter ϵ and the network structure parameter which represents fraction of excitatory cells. Their results showed that the network is capable of producing stationary bursting when ϵ is 1.0Hz or 1.9Hz, with the 90% or 98% excitatory cells. In this thesis work, one simulation ($\epsilon = 1.0$Hz and 90% excitatory cells) was chosen as a representative dataset for spatiotemporal data analysis and a second simulation ($\epsilon = 1.9$Hz and 98% excitatory cells) was used for validating the analysis results.

[1] In computational neuroscience, the electrical properties of neurons' membranes are depicted in terms of electrical circuits (termed membrane equivalent circuit accordingly), and Kirchoff's law describes the total current flowing across the neuron membrane as the sum of the capacitive current and the ionic currents.

These simulations were re-run on an 2.4GHz Intel Xeon E5-2620v3 system running Ubuntu Linux 16.04.3 using an NVIDIA Tesla K80 GPU with CUDA 8.0 libraries. Every spike produced during the simulation had its time (as an integer time step value with each time step being 0.1 ms,) and (x, y) neuron position recorded. There were 6×10^8 time steps in a simulation and it took about 120 hours to run to completion. The resulted datasets were about 30GB in size, stored in HDF5 data format.

The key information collected from simulation:

- **simulationEndTime** Total simulation time in seconds (60,000 s)
- **probedNeurons** Unique ID for each neuron ($0 \sim 9999$)
- **xloc** x location for each neuron ($0 \sim 99$)
- **yloc** y location for each neuron ($0 \sim 99$)
- **spikeshistory** Total spike count for every 10 ms time bins
- **spikesProbedNeuron** Spiking timing and location for every spike

Chapter 4

PRELIMINARY DATA ANALYSIS METHODS

Past neural spike data analyses have established knowledge about the major features of network behaviors. One of the most substantial network events is the convergence from spontaneous, random spiking to whole-network synchronous bursting, with individual bursts lasting for almost half a second and involving all or most neurons in the network firing multiple times. It is believed that besides spike rate and spike timing, network bursting is one of the mechanisms used in neural networks to carry significant information [58, 59]. Therefore, growing attention has been paid to study network bursts and it was chosen to be the main event to investigate in this thesis work.

It is crucial to have a preliminary analysis that helps us understand the data and sets the goals for the subsequent analysis. For this purpose, a set of methods were developed to identify and visualize different network events including neuronal avalanches and network bursting in order to learn about their characteristics. Many of the results from this step later served as input data for machine learning methods. In this chapter, data visualization and analysis methods were explained in detail; in addition, a data analysis workflow was presented at the end of this chapter to show the complexity of analyzing large datasets.

4.1 Avalanche Identification and Characterization

A neuronal avalanche is defined as a sequence of spikes that happen more frequently than the overall mean *inter-spike interval (ISI)*. From the literature, neuronal avalanches were identified by discarding the spatial information entirely, assimilating all neuron spikes into a single spike train [8, 9, 60, 12, 10]. In other words, any two consecutive spikes are grouped into the same avalanche if their ISI is smaller than the mean ISI. However, the original definition of perturbation propagation in self-organized criticality assumed that the perturbation is triggered externally and propagates through some neighborhood in a chain reaction [7]. From this definition, we concluded that spatial information should be also considered in identifying avalanche events.

Therefore, two different methods were used to identify neuronal avalanches. The first method followed the original neuronal avalanche definition by considering only the temporal information and grouping spikes as *temporal avalanches* [8]. In the second method, we took the original definition of self-organized criticality into account and applied both spatial and temporal constraints to identify groups of spikes as *spatiotemporal avalanches.*

4.1.1 Temporal Avalanches

From the simulation results, the **spikesProbedNeuron** dataset provided us with the information about the time and location of every single spike. This was then converted into the number of total spikes at each time step in order to get information about the *inter-spike interval (ISI)* in the spike data. ISI is defined as the time interval between two consecutive spikes in the spike train. Let N be the total number of spikes in the dataset, t_{n} be the occurrence time of the n_{th} spike, and T_{n} be the ISI given by:

$$T_n = t_{\mathrm{n+1}} - t_{\mathrm{n}}, \quad n = 1, 2, \ldots, \mathrm{N} - 1 \tag{4.1}$$

So, the mean ISI, $\overline{T}$, is

$$\overline{T} = \sum_{n=1}^{N-1} T_n \Big/ N - 1 \tag{4.2}$$

To identify temporal avalanches, we compared $\overline{ISI}$ with the intervals between consecutive time steps that have spiking activity. Due to the limitation on our temporal resolution 0.1 ms, there were situations in which one time step contained multiple spikes. In those cases, spikes fired at the same time step were grouped into the same avalanche.

4.1.2 Spatiotemporal Avalanches

We suspected that many temporal avalanches contained spikes that were close in time but distant in space and therefore should be considered as background activity and excluded from the avalanche event. The idea of the spatiotemporal method was to group spikes that were in close proximity in both space and time. This allowed us to exclude putative avalanches that were composed of a small number of spikes that were separated by large distances and clearly not causally related; it also helped us separately identify multiple avalanches in different parts of the network that overlapped only in time (but were spatially distinct). To do so, we defined a spatial constraint r as the radius for a circular spatial window to define a population of inter-connected neurons in which their spiking activities directly influence each other. Second, we determined a temporal constraint τ based on the number of neurons within the spatial window πr^2 and calculated τ as the mean ISI for this population:

$$\tau = \overline{T} \times \frac{M}{\pi r^2} \tag{4.3}$$

where M was the total number of neurons in the network (10,000).

Our purpose was to apply this spatiotemporal constraint to identify sequences of spikes that were close in both space and time by recognizing contiguous spiking activities. However, this method may encounter a situation where a spike in one avalanche could be spatially close to another avalanche at the same time. This happened when two or more avalanches that began separately but propagated towards each other, and there was no sure way to group these ambiguous spikes. Therefore, we decided to add the proposed spatiotemporal constraints on top of the avalanches that had already been identified using the temporal method. In this way, we solved the ambiguous spike problem because all of the avalanches were already pre-defined; what the spatiotemporal constraint did was throw out distant outliners from each avalanche.

We analyzed temporal avalanches from the last 1/4 of simulation because that was when the network had matured and reached a stable state as previously determined [56], and applied this spatiotemporal constraint only on avalanches that contained less than 10^4 spikes.

There were two reasons behind this decision: first, we found that there were two distinct groups of temporal avalanches identified and they showed very different network behaviors. The group with above 10^4 spikes was actually categorized as network bursts (as explained in section 4.2.1); in this analysis we focused on avalanches only and bursts were therefore disregarded. Second, this added constraint removed spikes that were considered irrelevant background activity or “noise” from temporal avalanches. Since most if not all neurons participated in synchronized spiking and generated more than 10^4 spikes in a single burst, noise only accounted for a very small fraction of the total number of spikes and would not cause a major impact in this analysis.

4.1.3 Self-Organized Criticality

The avalanches were examined on their characteristics such as size, duration, and their probability distribution. We defined avalanche *size* as the number of spikes, and *duration* as the time interval from the earliest spike to the last. In either case, avalanches were considered as propagation of system perturbations and their probability distributions were of interest. The size probability distribution was given by counting the occurrence of each possible size (range from 2 to the largest avalanche found) and plotting that as a fraction of the total number of avalanches versus size. We wanted to see if the avalanches obeyed the power-law relationship observed in self-organized systems and compare our result with the literature.

4.2 Burst Identification and Characterization

Two methods were used to identify bursts from the spike data. First, burst events were identified from avalanches by size. A second method applied a burst threshold on binned spike counts from the whole network as in [56], which was also consistent with methods in the physiological literature regarding bursting in cortical cultures. We investigated to see if these two different methods identified the same bursts in the network. To distinguish, we named them *avalanche bursts* and *binned bursts*, respectively.

4.2.1 Avalanche Bursts

Avalanches identified in the previous section represented spike events that were either temporally clustered or spatiotemporally clustered. Network bursts observed in living systems showed that bursts are synchronous spiking which are spatially distributed throughout the network over a period of time; therefore, bursts are essentially large avalanches. Results in [56] showed that all bursts contained more than 3×10^4 spikes. Having this size reference, we categorized all avalanches with size more than 10^4 as avalanche bursts.

4.2.2 Binned Bursts

As high-frequency occurrences of multiple spikes, network bursts can be observed from the elevated spike count over a period of time. Burst identification in [56] applied a burst threshold over binned network spike counts to identify burst start and end time (in time bins). They divided the spike data into 10 ms time bins (each bin was 100 time steps with every time step being 0.1 ms), calculated the number of spikes in each bin, and applied the burst threshold of 0.5 $spikes/sec/neuron$ or 50 $spikes/bin$. In our work, we used the same bin size and burst threshold to identify binned bursts.

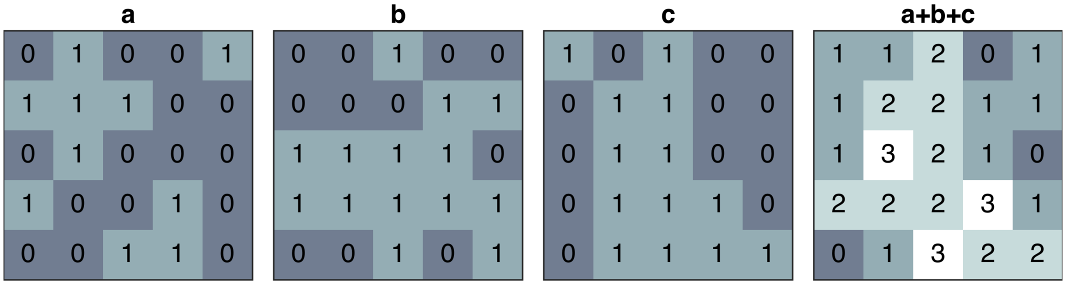
4.2.3 Burst Visualization

To apply spatiotemporal data analysis to bursts, we first performed visualization to provide a general idea of bursts' spatiotemporal patterns and properties. Intuitively, we discovered that by making a movie where each frame shows the (x, y) location of a spike in the network for a time step allowed us to see every spike in bursts. However, it is not practical to represent a time step for a frame when a burst happens over a couple thousand time steps. For instance, for a burst that happened across 3000 time steps, we are looking at a movie of a few minutes with reasonable frame rates[1]. This would work if we were to investigate a few

[1] The American standard for video frame rate is 30 frames per second (fps), higher quality video can have higher fps. In burst movies, low frame rates should be used in order to observe the spike activity in each frame.

bursts in details but not when thousands of bursts are identified in a simulation and the goal is to understand bursting behaviors in the big picture.

With the start and end time (in time bins) of bursts identified in the binned burst method, we applied *spike train binning* (Figure 4.1) with the same bin size (100 time steps). For each burst, images were created for each 10 ms bin instead of each time step, and the color of each square pixel in the 100×100 arrangement was determined by the number of spikes produced by the neuron at that (x, y) location. These images were either output individually or concatenated to produce burst movies.



a

0	1	0	0	1
1	1	1	0	0
0	1	0	0	0
1	0	0	1	0
0	0	1	1	0

b

0	0	1	0	0
0	0	0	1	1
1	1	1	1	0
1	1	1	1	1
0	0	1	0	1

c

1	0	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	1	0
0	1	1	1	1

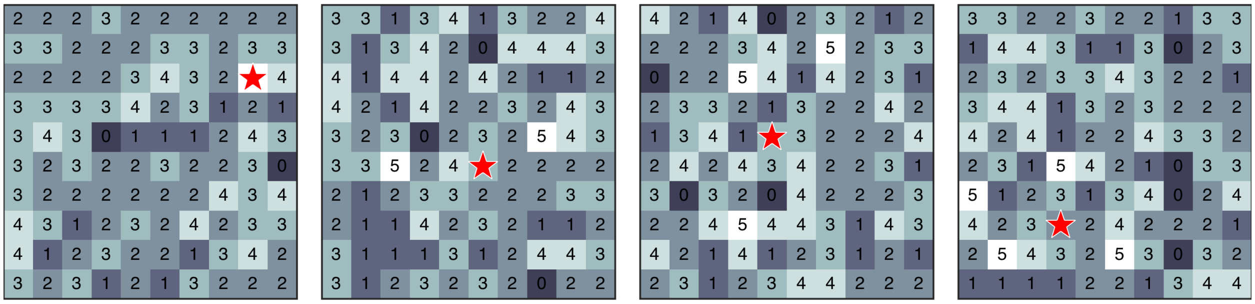
a+b+c

1	1	2	0	1
1	2	2	1	1
1	3	2	1	0
2	2	2	3	1
0	1	3	2	2

Figure 4.1: Example of spike train binning (bin size = 3). Image a, b and c represent three consecutive time steps in a 5×5 arrangement. The binary values in each location represents a single spike train value. The rightmost image shows the result of binned spike trains of a, b, and c. The color in each square corresponds to the spike count or spike rate of the neuron at that (x, y) location. The brighter colors indicate higher spike rates.

4.2.4 Burst Origin

Burst visualization revealed that bursts originated in small spatial regions and then propagated across the network. Each burst initiation was then analyzed to assign a (x, y) location as its *burst origin*. From the previous section, images were made for each time bin in a burst using spike train binning. The first image of a burst (corresponding to the spikes occurring in the first 10 ms of that burst) contained information about its initiation, and was used to find the centroid coordinates of locations with the highest spike rate, or brightest pixel values. In other words, the origin was computed as the mean of the x and y coordinates of neurons that produced the most spikes in the first 10 ms; this was invariably either a single neuron or a compact cluster. We called this method *brightest pixel selection* and examples of identifying centroid using this method is shown in Figure 4.2.



2	2	2	3	2	2	2	2	2	2
3	3	2	2	2	3	3	2	3	3
2	2	2	2	3	4	3	2	★	4
3	3	3	3	4	2	3	1	2	1
3	4	3	0	1	1	1	2	4	3
3	2	3	2	2	3	2	2	3	0
3	2	2	2	2	2	2	4	3	4
4	3	1	2	3	2	4	2	3	3
4	1	2	3	2	2	1	3	4	2
3	2	3	1	2	1	3	2	2	2

3	3	1	3	4	1	3	2	2	4
3	1	3	4	2	0	4	4	4	3
4	1	4	4	2	4	2	1	1	2
4	2	1	4	2	2	3	2	4	3
3	2	3	0	2	3	2	5	4	3
3	3	5	2	4	★	2	2	2	2
2	1	2	3	3	2	2	2	3	3
2	1	1	4	2	3	2	1	1	2
3	1	1	1	3	1	2	4	4	3
3	1	2	3	2	3	2	0	2	2

4	2	1	4	0	2	3	2	1	2
2	2	2	3	4	2	5	2	3	3
0	2	2	5	4	1	4	2	3	1
2	3	3	2	1	3	2	2	4	2
1	3	4	1	★	3	2	2	2	4
2	4	2	4	3	4	2	2	3	1
3	0	3	2	0	4	2	2	2	3
2	2	4	5	4	4	3	1	4	3
4	2	1	4	1	2	3	1	2	1
2	3	1	2	3	4	4	2	2	2

3	3	2	2	3	2	2	1	3	3
1	4	4	3	1	1	3	0	2	3
2	3	2	3	3	4	3	2	2	1
3	4	4	1	3	2	3	2	2	2
4	2	4	1	2	2	4	3	3	2
2	3	1	5	4	2	1	0	3	3
5	1	2	3	1	3	4	0	2	4
4	2	3	★	2	4	2	2	2	1
2	5	4	3	2	5	3	0	3	2
1	1	1	1	2	2	1	3	4	4

Figure 4.2: Examples of burst origin by computing the centroid of the brightest pixels. For each image, numbers in each square represent pixel values and red pentagram marks the centroid of the brightest pixels (highest pixel value = 5). Images from left to right show the centroids of 1, 2, 3, and 4 brightest pixels, respectively.

When selecting the brightest pixel, a threshold of 2 was used to avoid situations in which a bin only contains neurons that fired once and their centroid did not approximate burst origin. This is because time bins are determined sequentially from the very first time step to the last, by applying the binned burst method and aggregating 100 time steps into one, the burst can initiate at any time in the first bin. If the burst starts late in the bin, the highest spike count in that first bin can be as low as 1. As we know that the burst threshold is 50 *spikes/bin*, this means there were more than 50 neurons that had the same spike count. When this happens, the mean position of these neurons can significantly deviate from the origin location that we anticipated using this method. Therefore, we applied a threshold of 2 in selecting the brightest pixel and if the condition was not met, the next bin was used to calculate the origin. These computed origins were manually examined to double-check that there were no unreasonable artifacts.

In the burst events we identified, there were cases when two bursts happened in tandem, originated at different locations in the network but were temporally inseparable; they propagated towards each other and eventually collided and formed a greater burst until it exited the network, we called them *twin bursts.* In cases of a twin burst, the origin of the burst that happened slightly earlier was identified as the origin.

4.2.5 Burst Propagation Speed

Since bursts propagate outward from their origins and across the entire network as a wave, their propagation speeds can be determined by using the origin location and brightest pixel selection method. For each burst, we searched for the brightest pixels for each image (corresponding to the spikes occurring in the each 10ms bin in the burst) and their distances to the origin were then calculated (and averaged if there were more than one brightest pixel in the image). Because each image represents a time bin, the time differences for every bin relative to the first bin were known. By performing the same calculation for every image in a burst, we had the relative distance and time (bin difference) for any bin to compute the propagation speed. The burst speed V for each bin was given by

$$V_{\mathrm{ij}} = \frac{d_{\mathrm{ij}}}{t_{\mathrm{ij}}} \quad (units : ms^{-1}) \tag{4.4}$$

where i is the burst number, j is the bin number, d_{ij} is the distance (or mean distance) between the (x, y) location of the brightest pixel (or pixels) in bin j and the origin location of burst i. t_{ij} is the time difference of bin j relative to the first bin in burst i in milliseconds. As the spatial grid used in the simulation is dimensionless, speed is presented in units of ms^{-1}.

For every burst, we calculated the speed for every bin relative to the first bin and finally these speeds were averaged to produce the overall *burst propagation speed.* To avoid situations as discussed in the previous section where the brightest pixel being 1 and introduce inaccurate results, we discarded the information from the first and last two bins for every burst.

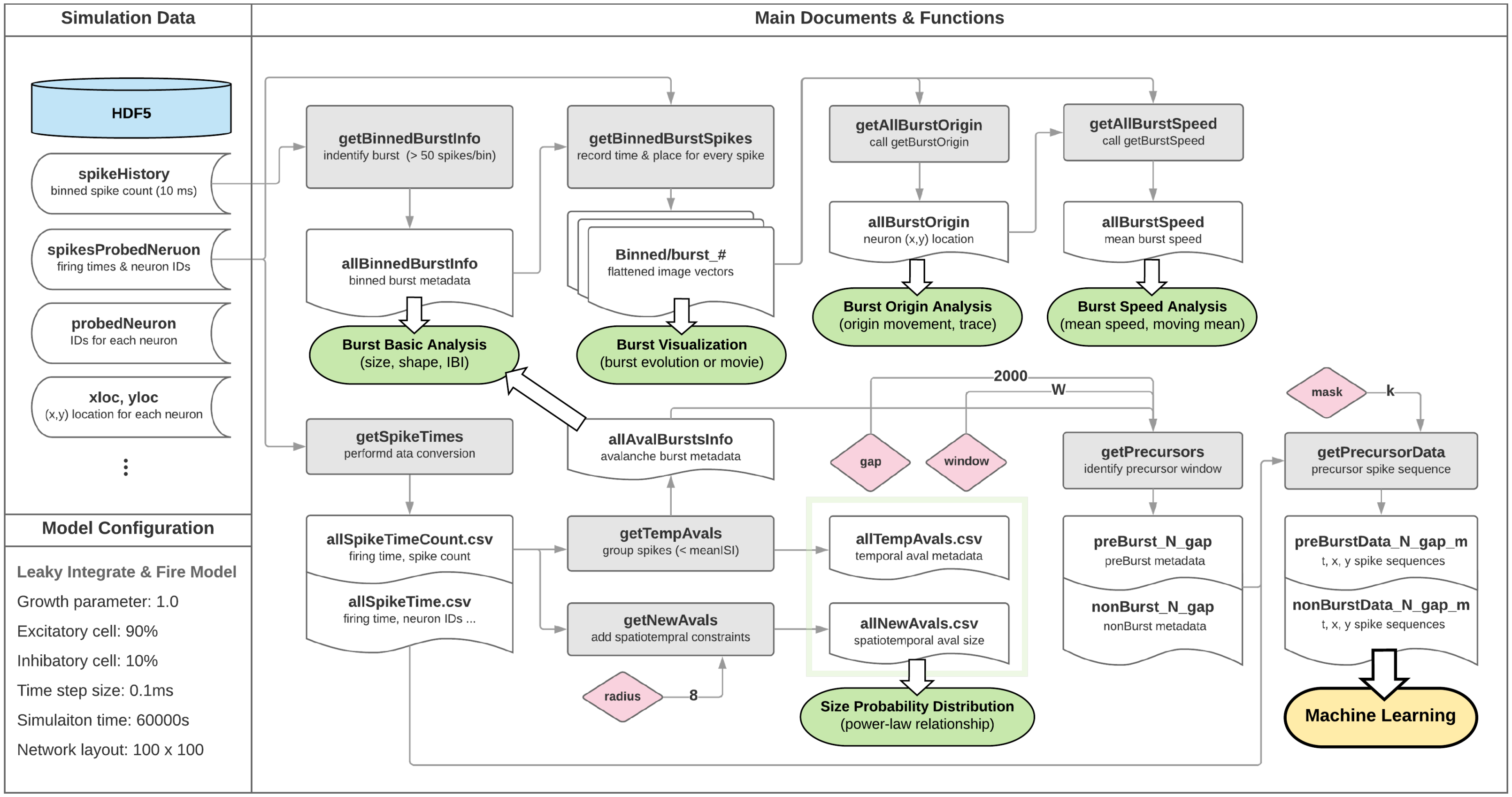


Figure 4.3: High level workflow diagram of preliminary analysis. This is a simplified diagram that left out a great amount of details intended to serve as an overview of preliminary analysis as well as a demonstration of the complexity of the work. Blue and white entities represents datasets or data documents, grey entities are major functions and methods, pink entities are important self-defined variables used in this work, and green entities represent the analysis results.

Chapter 5

MACHINE LEARNING METHODS

In the previous chapter, we conducted a number of visualizations and investigations of macroscopic, whole-network behaviors on our dataset; we identified avalanches and bursts from the network and examined important characteristics of these behaviors to compare with the literature. Thus far, we have performed a hypothesis-driven approach in which we defined the hypothesis and attempted to provide answers to the specific questions for the hypothesis. However, data can contain information that is undiscovered and therefore cannot be completely hypothesized and analyzed. With the high dimensionality, complexity, and volume of our dataset, we believe that machine learning can enable us to discover unfathomable information or patterns, from which we can learn much more than through our current graphs and images.

Machine Learning (ML) methods are computer algorithms capable of adaptively improving their performance of a task based on their own previous experience and discovering hidden patterns in high dimensional data [61, 62, 63]. These methods pertain to the ability of data-driven approaches to abstract and gain information about a system directly from observed data without establishing mechanistic relationships that govern the system. Due to these advantages that ML has to offer which align well with our own interests and the idiosyncrasies of the dataset, we decided to utilize ML for subsequent analysis. Based on the preliminary results, we further applied a set of ML techniques to study the burst spatiotemporal behaviors. Our goal was to perform relatively straightforward ML methods to see if a small number of spatiotemporal spikes were predictive of the macroscopic network bursts and therefore could be identified as trigger events from the large amount of spike data. If so, it would suggest that these microscopic patterns were behaviorally significant.

Figure 5.1 shows an overall data analysis process which illustrates the key steps from collecting data to discovering useful information. The previous chapters provide details about step 1 and 2 in Figure 5.1; data preparation procedures, ML algorithms, and performance metrics used in step 3, 4, 5 are explained in this chapter.

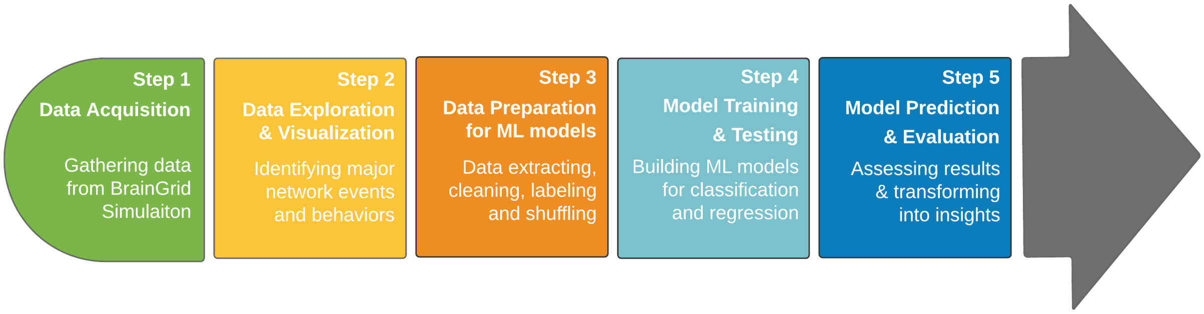

Step 1
Data Acquisition
Gathering data from BrainGrid Simulaiton
Step 2
Data Exploration & Visualization
Identifying major network events and behaviors
Step 3
Data Preparation for ML models
Data extracting, cleaning, labeling and shuffling
Step 4
Model Training & Testing
Building ML models for classification and regression
Step 5
Model Prediction & Evaluation
Assessing results & transforming into insights

Figure 5.1: Overall data analysis process.

5.1 *Data Preparation*

Prior to any direct application of ML algorithms, it is important to be conscious of the content of the raw data so essential pre-process procedures can be taken to ensure the data quality and effectiveness of the algorithms. In this case, because every spike produced by the network was recorded, there was no missing values or inconsistent features, but decisions about how much data needed (among hundreds of millions of spikes) to predict each burst and how to represent spatiotemporal spikes must be made. The quality, amount, and selection of data are critical to the success of ML solutions and human attention with in-depth understanding about the dataset were required for decision making. The data preparation task is subdivided as a set of relevant steps and explained in the following subsections.

5.1.1 Data Extraction: Pre-Burst & Non-Burst Precursors

To investigate if particular patterns of spatiotemporal activity triggered bursts, we isolated spike sequences just before bursts as *pre-burst precursors*, and spike sequences temporally distant from bursts as *non-burst precursors*. These precursor data were used as input for ML algorithms. To find the target data sequences, the start and end times of bursts, or *burst boundaries*, defined in time steps, were needed. Our two burst identification methods picked out the same bursts, but avalanche bursts were grouped in time steps whereas binned bursts were in 10 ms time bins. Henceforth, the burst boundaries determined by avalanche bursts were used in extracting the sequences for precursors.

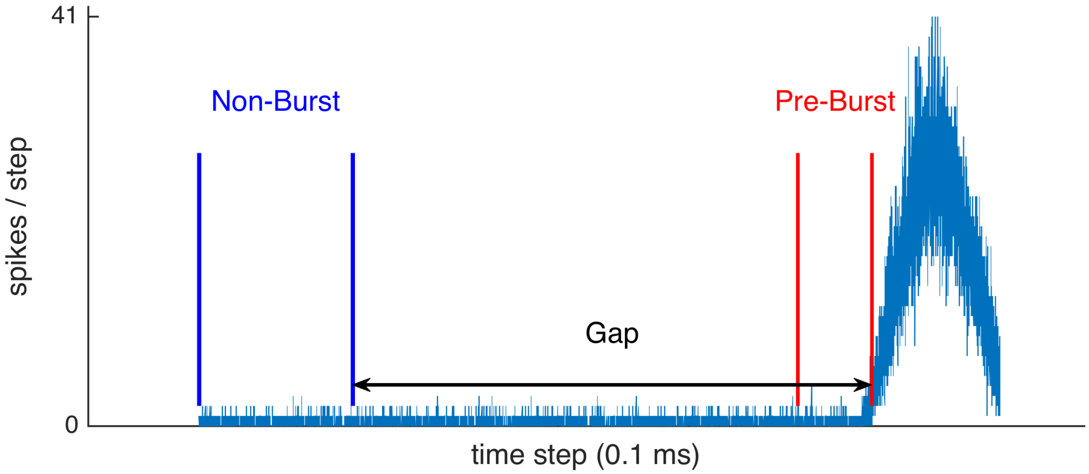


Figure 5.2: Definition of non-burst and pre-burst precursor windows ($W = 500$). The spike activity for every time step before and during a burst is plotted showing the highest spike count of 41. Blue and red lines indicate the time windows for non-burst and pre-burst precursors. A gap of 2000 spikes was used to define the end boundary of non-burst window.

Figure 5.2 illustrates how the time windows of pre-burst and non-burst precursors were defined relative to a burst event. For each avalanche bursts, we selected W consecutive spikes right before each burst as a pre-burst data sample and selected another W spikes with a distance of 2000 spikes (the“gap” in Figure 5.2) before each burst as a non-burst data sample.

The gap was chosen based on the *inter-burst intervals (IBIs)* and *mean spike rate* calculated from previous analysis. The smallest IBI found in the data was 5445 time steps and the mean spike rate was 0.9503 *spikes/step*. With these information, we assumed the burst boundaries were identified accurately and there was no more than 1 spike at each time step when the network was not bursting; a gap of 2000 spikes gave us sequences that were far enough from the upcoming bursts, eliminating the chances of precursor window containing any pre-burst activities, and at the same time distant from the preceding bursts to avoid including activities that may pose post-burst behaviors. Additionally, we investigated the location of the spikes that best predict burst initiation (those spikes might form patterns that “trigger” bursts) in the data sequence. We did this by varying the size of the precursor window and the “mask” (defined in Figure 5.3). Our purpose was to find the optimal pre-burst window which is the smallest (contains as few spikes as possible) and farthest (as far away from the burst as possible) yet still highly predictive of bursts.

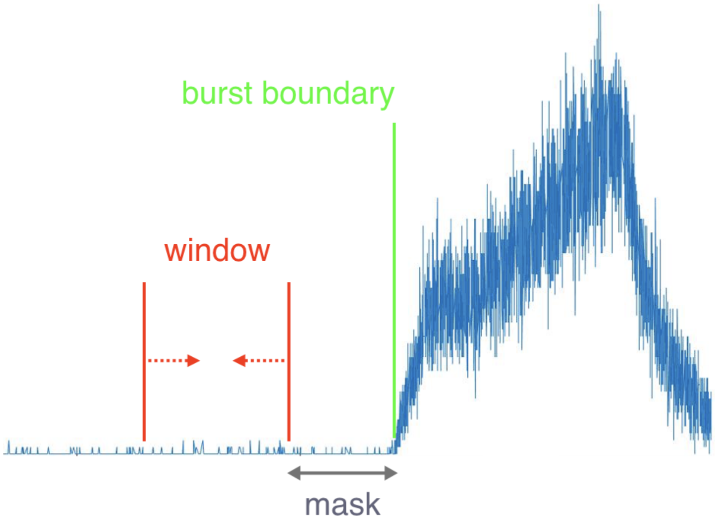


Figure 5.3: Illustration of the optimal pre-burst window and mask. Green line corresponds to the burst start boundary defined in avalanche bursts. Red window represents pre-burst window which contains W spikes, and mask contains k in between the window and the burst. The smallest and most predictive pre-burst precursor can be found by reducing window size W and increasing mask size k.

In this work, we experimented various W values, ranging from 5 to 500. For each spike i in a data sample, we retrieved its neuron (x, y) location and its firing time, $\phi_i = t_i - t_0$, relative to the first spike (spike 0) in that data sample. Every data sample was then arranged in the following format to include its spatial and temporal information:

$$\phi_0, x_0, y_0, \phi_1, x_1, y_1, \ldots, \phi_{W-1}, x_{W-1}, y_{W-1} \tag{5.1}$$

While spatiotemporal datasets are mostly represented as images, it often requires the redesign of most machine learning methods to accept image or video input. This alternative and yet simple representation reduced the data complexity by treating each dimension (ϕ, x, y) of a spike as a feature. For a sequence of W spikes, there are $3 \times W$ features in each data sample.

5.1.2 Data Labeling: Ground Truth

To study how the microscopic activity affect network bursts, we performed several *binary classification* and *multivariate regression* tasks. First, we investigated if the pre-burst precursors were predictive of a burst compared to non-burst precursors. Each pre-burst precursor was labeled as 1 and each non-burst precursor as 0 for binary classification. If precursors were capable of predicting burst initiation, we wanted to see if burst origin locations were predictable by labeling each pre-burst precursor with its subsequent burst origin (x, y) for multivariate regression analysis.

5.1.3 Data Cleaning & Shuffling

In our data samples, there were cases that avalanche method was not able to identify the burst boundaries as we expected and further contaminated the result of precursor data extraction. In these cases, many of the bursts had their start boundaries identified too late or end boundaries identified too early because there were times when no spike at a time step was recorded. These “drop downs” in spike count indicated spike intervals that were bigger than the mean ISI; the drop downs then became the cutting points in grouping avalanches and the burst boundaries. When this happened, the precursor windows that were determined entirely on burst boundaries would not be appropriate anymore (Figure 5.4).

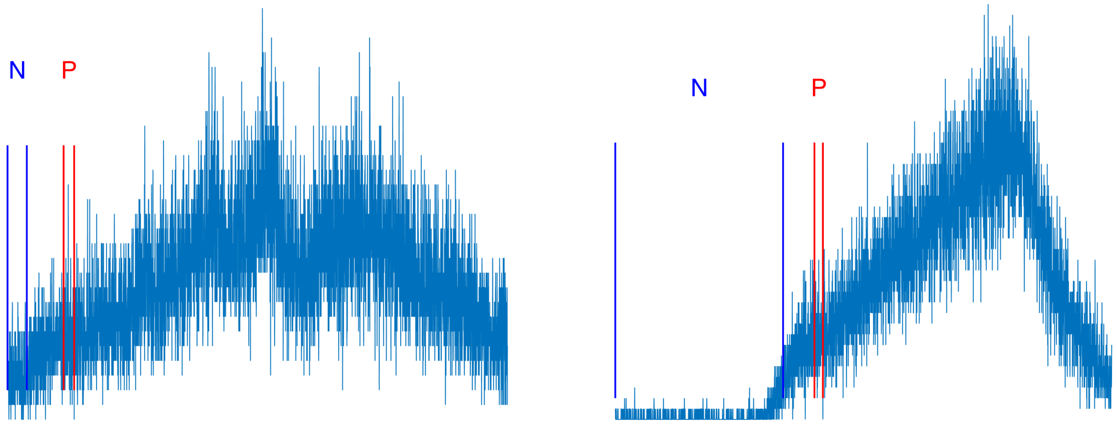


Figure 5.4: Examples of bad precursors as a result of poorly identified burst boundaries. The blue window "N" represents the non-burst precursor; the red window "P" represents the pre-burst precursor.

We speculated that these poorly identified burst boundaries could disappear as we improved the time resolution in the simulation. With finer time resolution, ideally having only one spike for each time step, we should have a better estimation of mean ISI that were able to group spikes into different events more accurately. However, finer resolution translated to longer simulation and larger dataset. At this point, we decided to work with what we had and performed data cleaning to remove the bad precursors from our data samples.

Data samples with either a bad precursor window (width smaller than window size N) or a bad gap (width with smaller than gap size) were found in our data. Among 9729 bursts, there were 538 of them found to produce "bad data" (bad pre-burst or bad non-burst precursor window) and they happened throughout the data, showing no particular occurrence pattern. We manually inspected some of the precursors and decided that these thresholds were quite effective on finding the bad data while keeping the good ones. Bad data samples were therefore removed from our dataset. After data cleaning, we were left with precursor sequences from 9191 bursts as input for ML methods.

On the other hand, one important macroscopic feature of the simulation behavior was that bursts settled down to a few stable burst initiation locations during development and the majority of bursts originated from those regions at later stages of network development. Including all the data samples produces an imbalanced dataset and can skew the result. We wanted to apply ML techniques to activity patterns that were less easily predictable, so we chose only the first 500 bursts — those with origins that varied across the network — to analyze. As a control, we applied the same regression analysis to non-burst data samples. Moreover, to prevent prediction results being affected by sequential temporal relationships between consecutive bursts or between consecutive data samples, all data samples were randomly shuffled before they were used for model training.

5.2 Binary Classification

Binary Classification is a supervised learning method of classifying a data into two predefined categories or labels, in our case, “no burst” and “burst”. Among various ML classifiers, Decision Tree (DT) and Support Vector Machine (SVM) with both linear and nonlinear kernels were chosen to predict burst initiation.

DT uses a tree-like graph for decision making; it constructs the tree using the top-down approach by considering information gain as a criterion and chooses the best feature to split each node so that it produces the “purest” subsets and stops when data cannot be split further. In other words, a decision tree is built by calculating feature importance. DT is easy to interpret by non-statistician and is intuitive to follow, it copes with irrelevant features and is able to combine heterogeneous data types into a single model. As for SVM, it finds the maximum margin hyper-plane that best separates two classes in a high-dimensional feature space by plotting each data sample as a point in n-dimensional space (where n is the number of features in the data sample) with the value of each feature being the value of a particular coordinate. SVMs are very universal learners, they can provide generalized models in the presence of many features as long as the data is separable with a wide margin. In its basic form, SVM learns linear threshold function, but with different kernel functions, SVM can also be used to learn polynomial classifiers. While SVM is well suited to handle high dimensional data, DT can be a cheaper, yet effective solution if the majority of the features do not contribute much information to the classification problem.

Classification models were confirmed using k-fold cross validation ($k = 10$) in which the data was divided into k subsets and every time only $k - 1$ subsets were used to train the model, leaving the last subset for testing. The performance of the classification model is evaluated in terms of accuracy, precision, recall, confusion matrix, and F1 score. Since F1 score provides a harmonic mean of precision and recall by combining the two numbers into one measure, it is chosen to be the main performance metric shown in the results.

5.3 Multivariate Regression

Multivariate regression is a technique that estimates a single regression model with more than one outcome variable. In our case, there are two outcome variables to predict: x and y location of the burst origin. Linear regression, Lasso regression, Ridge regression and artificial neural networks (ANNs) were used for predictive modeling.

Linear regression is a linear approach for modeling the relationship between explanatory predictors (or independent variables) and scalar responses (or dependent variables). Simply put, linear regression model gives a linear equation that best describes the relationship between features and label by minimizing the sum of the squared error. Lasso and Ridge regressions are extensions of linear regression with different regularization methods to reduce model complexity and overcome overfitting. Lasso and Ridge regression are closely related, but only Lasso has the ability to select predictors. In this work, ANNs were also used for regression modeling since ANNs use adaptive weights and can approximate non-linear functions of their inputs, they can easily adapt to different problems by customization; thus, ANN become a great alternative when a more traditional model cannot fit the solution. On the other hand, the customization also greatly effect the model performance and model tuning is required. We trained the ANN models with *grid search* for hyperparameter optimization; a set of different hyperparameters in Table 5.1 were experimented to find the best combination.

Regression model accuracy were assessed by R-squared score, mean absolute error (MAE) and root mean squared error (RMSE). While R-squared allows us to understand the percentage of variance in the target data explained by the model, MAE and RMSE help quantify the error by averaging the residuals of the model and provide different insights of the result.

Table 5.1: ANN Model Hyperparameters

Hyperparameter	**Types/Values**
activation function	logistic, tanh, relu
learning rate	constant, invscaling, adaptive
solver	sgd, adam
# of hidden layers	1, 2, 3, 4
# of neurons per layer	20, 100, 200, 500, 1000
batch size	10, 50, 100, 250, 500
max iteration	100, 200, 500, 1000, 2000

Chapter 6

RESULTS

6.1 Avalanche Characteristics

Our simulation data contained a total of 570,189,562 spikes and 19,183,767 temporal avalanches with size ranging from 2 to 55,205 spikes. Among them, 3,199,919 temporal avalanches were identified for the last 1/4 of the simulation (corresponding to the last 7 of 28 days *in vitro*) when the network was stationary as previously determined [56]. After applying spatiotemporal constraints, the number of avalanches was eliminated down to 156,604. The size probability distribution in Figure 6.1 shows that there were clearly two groups in the data; one group has sizes ranged from 2 to 10^3, another with size between 10^4 to 10^5.

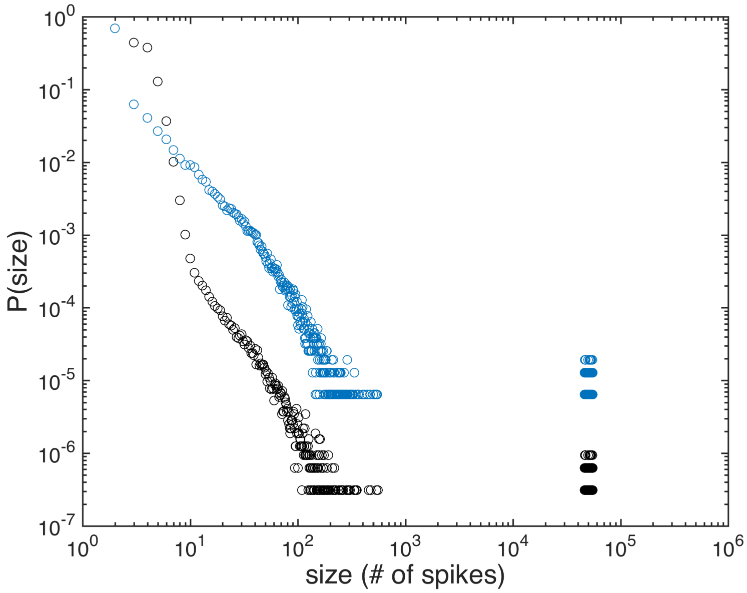


Figure 6.1: The probability distribution of avalanche size for the last 1/4 of data. Black circles represent temporal avalanches and blue circles represent spatiotemporal avalanches.

6.2 Burst Characteristics

After bursts were identified, we gathered information about the size (number of spikes), width (time steps involved), mean spike rate (size/width), peak (the time bin that contained the highest number of spikes) for each burst and they were consistent with the results in [56]. Furthermore, we generated images to represent each bin in a burst and identified the origin and propagation speed for each burst. The results are presented in the following sections.

6.2.1 Burst Spatiotemporal Patterns

The network was visualized as sequence of binned images (or movie frames) with each pixel as a neuron and the pixel color corresponding to spiking rate to display the spatiotemporal pattern of bursts. Figure 6.2 presents an example visualization of a whole network burst as it develops over time. The first image of the burst shows its origin location as frequent spike activities within a small region, and the evolution development reveals that bursts originate in a single location and spread as a wave of activity across the network.

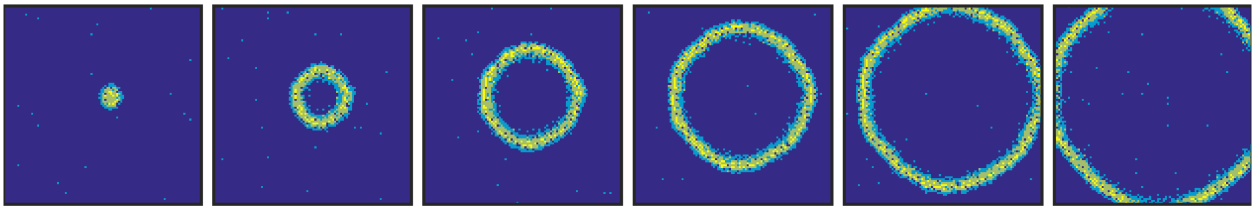


Figure 6.2: Spatiotemporal evolution of a single burst (from left to right). Each image includes 10ms (100 time steps) of activity and images are 30 ms (300 time steps) apart.

6.2.2 Burst Origin Evolution

Figure 6.3 displays the evolution of burst origin locations for all 9729 bursts. It shows that burst origins varied widely in early network development and clearly settled down to a small number of origins that occurred repeatedly in a random order, or “active origins”, and these active origins changed to a different set of locations over time. Based on these insights, we further examined the burst origin occurrence patterns in different stages of the network development and displayed them in Figure 6.4.

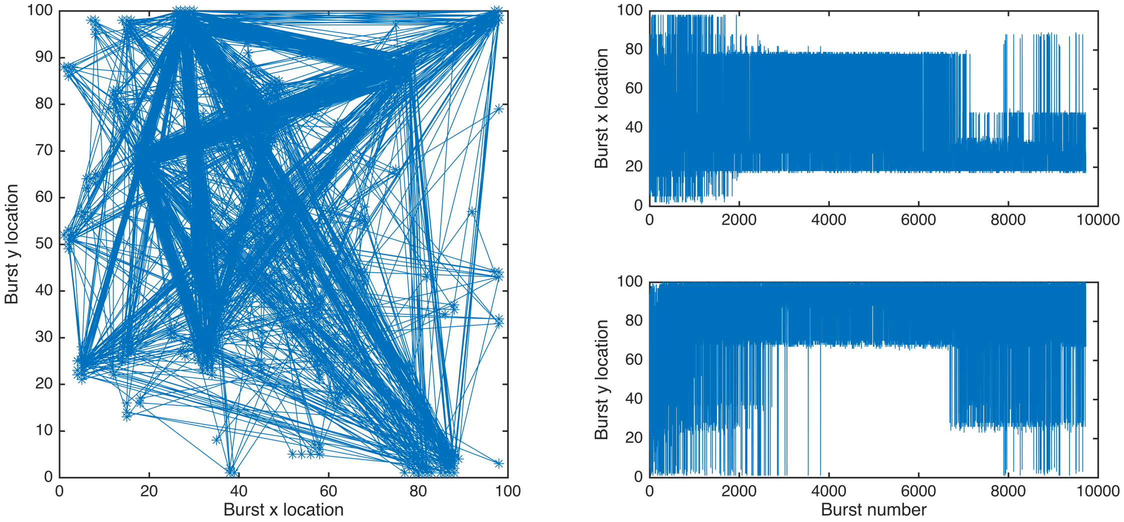


Figure 6.3: Evolution of burst origin locations. **Left**: (x, y) location of each burst origin (*) and their sequence (lines). **Right**: x (top) and y (bottom) coordinates of burst origins plotted versus burst number which also indicates the time of occurrence of each burst.

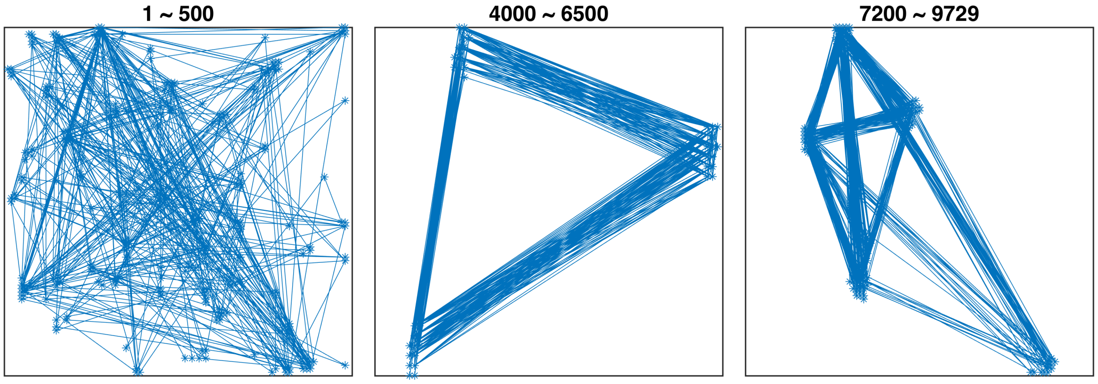


Figure 6.4: Active origins in different stages of development. Bursts that were included in each stage are listed on top of each image with their burst numbers.

6.2.3 Burst Speed Evolution

Figure 6.5 shows burst propagation speed as a function of burst number for all bursts, along with a 100-burst moving average. Consistent with the results in [56], burst initially propagates slower, starts out with a speed of 0.4 ms^{-1}, and rapidly speed up early on (burst 1 $\sim$ 2000); speed continues to grow with gradual progression then drop and raise again right before it enters the last 1/4 of simulation (around burst 6500). After the network reaches maturity (burst 8000 $\sim$ 9729), burst speed settles to a stationary speed around 0.85 ms^{-1}.

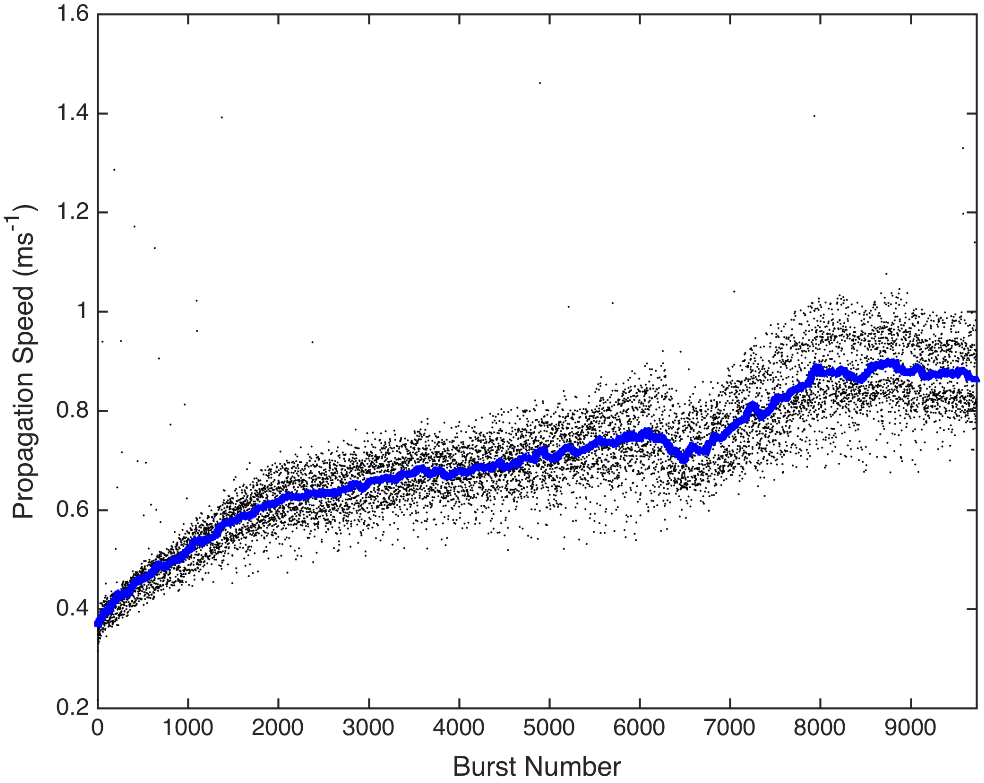


Figure 6.5: Evolution of burst propagation speed. The black dots mark the burst speed during network development for individual bursts and the blue line shows the moving average of 100 bursts. There is a clear trend showing bursts increase in propagation speed over time.

6.3 Classification: Burst Initiation

Binary classification results with different window size W are shown in Table 6.1. We found that DT and linear SVM both had the highest F1 score (0.9992 and 0.9983) when $W = 10$, and also produced the smallest 10-fold error (0.12% and 0.16% misclassification rate). Similarly, both methods performed the worst when $W = 500$, linear SVM in particular, its F1 score plummeted from above 0.99 to 0.88. DT models performed well in all scenarios and took notably less time to train. The polynomial SVM model did much worse than the other two models in which its F1 score only achieved 0.90 when $W = 10$ and 50.

Table 6.1: Classification Model Performance

Method	W	F1 score	10-fold error	training time (s)
Decision Tree	5	0.9859	0.0 F1 score116	0.0477
	10	0.9992	0.0012	0.0304
	50	0.9967	0.0031	0.1360
	100	0.9951	0.0052	0.2489
	500	0.9803	0.0201	2.4913
Linear SVM	5	0.9905	0.0109	366.1565
	10	0.9983	0.0016	437.9697
	50	0.9981	0.0019	4.7366
	100	0.9973	0.0028	345.4788
	500	0.8809	0.1793	384.0659
Polynomial SVM (d=2)	5	0.7573	0.2064	335.8493
	10	0.9298	0.2017	404.4486
	50	0.9501	0.0809	417.7756
	100	0.8277	0.1797	391.9752
	500	0.8480	0.1780	420.6529

With the size of window and mask (defined in Figure 5.3) incrementing from 1 to 50 spikes, we ran 2500 DT models (50 × 50) to find the location of burst trigger pattern and the result is shown in Figure 6.6. It is clear that the F1 scores were quite consistent as long as the window contains more than 5 spikes; there was a very subtle trend showing that the performance drops with increased window size. On the other hand, model performance is closely correlated with the mask size: the larger the mask, the poorer the performance. The best F1 scores happen when window contains 8 to 30 spikes and mask has fewer than 5 spikes.

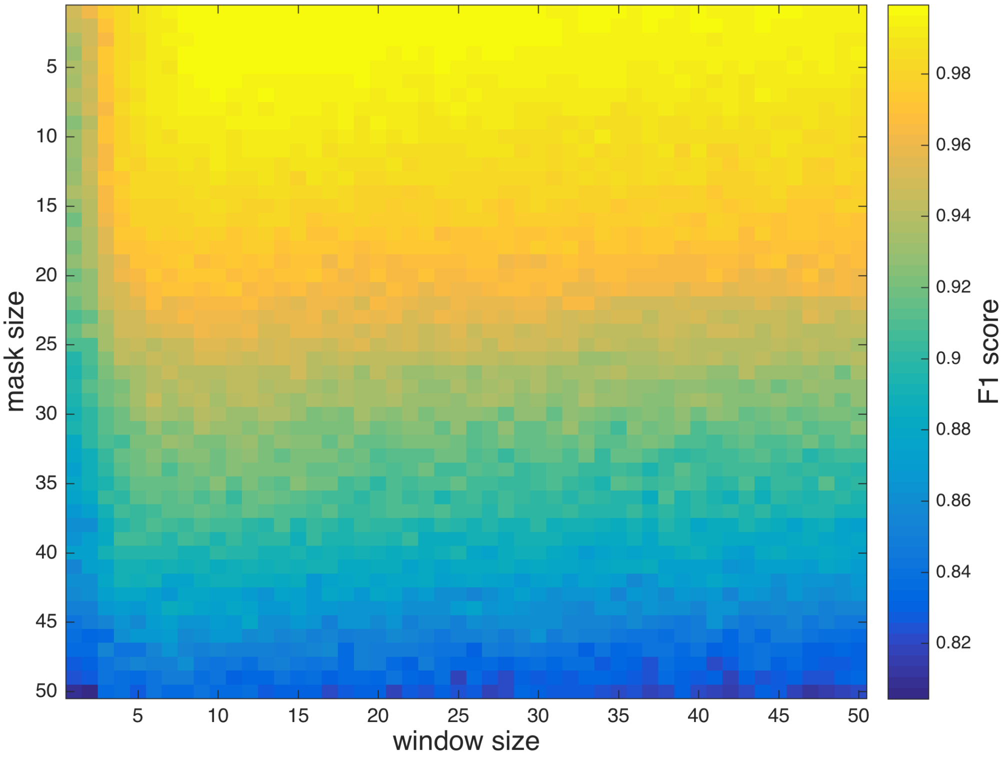


Figure 6.6: Classification result for different window and mask size varying from 1 to 50 spikes. Each color in the square corresponding to the F1 score for the particular window and mask combination at that (x, y) location.

6.4 Regression: Burst Origin Prediciton

Regression results on pre-burst data are shown in Table 6.2. The result of grid search showed that the best performing ANN model had 2 hidden layers, each with 1000 neurons with ReLU activation functions, constant learning rate, adam solver. The best batch size is 50 and maximum iteration is 200. Regression results from all models achieved above 95% R2 score. The control, non-burst data samples were not predictive of burst origin with poor R2 scores and large MAE and RMSE ($30 \sim 50$); therefore, the results are not shown.

Table 6.2: Regression Model Peformance

Method	W	R2	MAE	RMSE
Linear	10	0.9579	4.2872	5.3510
	50	0.9923	1.8264	2.4884
	100	0.9853	2.6417	3.4106
Ridge	10	0.9586	4.2707	5.2998
	50	0.9924	1.8190	2.4817
	100	0.9877	2.4233	3.1111
Lasso	10	0.9615	3.4920	5.1873
	50	0.9943	1.5071	2.1304
	100	0.9947	1.4771	2.0244
ANN	10	0.9552	3.4971	5.613
	50	0.9949	1.4649	2.0133
	100	0.9920	1.7840	2.4784

Chapter 7
DISCUSSION

The probability distribution of avalanche size in Figure 6.1 displays that the spatiotemporal avalanches follow the power-law relationship more consistently especially when compared to small temporal avalanches (size smaller than 50). This suggests that small temporal avalanches are over-represented, violating the power law distribution; we hypothesized this could have been caused by "spurious" avalanches: spikes occurring near each other in time but far away in space. Our spatiotemporal constraint eliminated some spurious avalanches, either by reducing their size or removing them entirely. The result was a relationship that is more consistent with power law distribution, confirming the hypotheses.

For burst analysis, Figure 6.3 and Figure 6.4 show that the network develops a set of active origins that change over time, and bursts become highly likely to initiate at these locations as the network development progresses. Figure 6.5 reported a trend that burst propagation speed increases during development and is consistent with the literature [64].

From the classification results, high F1 scores from the majority of our models showed that pre-burst precursors were very predictive of burst initiation. For each of the three methods, $W = 10$ and $W = 50$ yielded the highest F1 scores among all models indicating the most predictive spikes are within this range of window sizes. By comparison, polynomial SVM models performed much worse than linear kernels, suggesting that the feature space of our data is linearly separable and that the polynomial kernel may have been over-fitting. The results from polynomial SVM models are therefore considered non-representative and disregarded in the following discussions.

Overall, DT and linear SVM models had very competitive results; however, it took significantly less time to train DT models than SVM models despite different feature sizes

$(3 \times W)$. It is quite noticeable that SVM model has one exceptional training time of 4.7366 seconds when $W = 50$ while other models took more than 300 seconds to train. Regardless of the feature size, SVM will continue to map data into higher and higher dimensions until a hyperplane can be formed to segregate one class from another; this suggested that an optimal hyperplane was most obvious to identify when $W = 50$. As SVM takes all features of the data into account, the fact that the performance of SVM model achieved over 0.998 when $W = 50$ and $W = 100$ and only 0.8809 when $W = 500$ indicates that there was an increased amount of redundant features or "noise" when we included 500 spikes. Although SVM can be a powerful binary classifier for high dimensional data, the presence of irrelevant or redundant features in the dataset may provide repetitive or even contradictory information which gives rise to misleading the construction of a precise SVM model. This property of SVM helped us reduce the search window for burst triggers from the included 500 spikes down to 50, and at most 100 spikes

On the other hand, DT uses only a subset of the features to construct the model and all models reached high F1 scores (over 0.98), this indicates that there were some patterns predictive of a burst in all cases $(W = 5, 10, 50, 100, 500)$. Additionally, we found that when $W = 500$, there were only 82 out of 1500 predictors $(3 \times W)$ being used to construct the model and 35 of them were from the last 50 spikes in data sample; alghough there was no distinctive pattern showing any of the 3 types of feature (ϕ, x, y) to be more important than others or a consistent set of predictors found in every case. These results also suggested that DT can be used as the first stage of analysis for large quantities of biological data to identify regions of interest for subsequent examination. DT was thereby used to investigate the location of burst trigger pattern shown in Figure 6.6. The result shows that a burst is no longer predictable when mask size is bigger than 50 spikes if we consider F1 score smaller than 0.8 as poor prediction; this suggests that the trigger pattern can contain fewer than 10 spikes and that are most 50 spikes before the burst starts.

The multivariate regression models used here predicted two outcome variables representing the (x, y) location for burst origin based on pre-burst activity. From the R2 scores, one can see that at least 95% of the data variance can be explained by our models. When $W = 50$, all models reported the best R2 score of over 99.2% with small MAE and RMSE prediction errors (less than 2.5). Considering the fact that burst origin locations were approximations, it is quite accurate that the predicted (x, y) locations were only off by 1 to 3 neuron positions in a 100 × 100 network setting. Unlike classification results which the best scores were achieved for both DT and SVM when $W = 10$, regression results report the worst R2 scores and this indicates that longer sequences of spikes are needed to improve burst origin prediction.

For models of $W = 50$, lasso model performed better (99.47%) than linear and ridge models, suggesting that only a subset of features were important for burst origin prediction since lasso selects a subset of features by reducing the coefficients of others to zero. ANN models are notoriously sensitive to the fine-tuning of hyperparameters and thus the result can vary greatly. The best R2 score we achieved is 99.49% while continued experimentation might lead to better performances. For all of our regression models, RMSE was a little higher than MAE. Since RMSE weights large errors more highly than MAE, this indicates that there was a small number of distant outliers in our prediction.

Chapter 8
CONCLUSION

This thesis investigated the spatiotemporal behaviors of full spiking activity from a closed-loop simulation of neuron-activity-driven network development which used the simplest model of cortical culture growth to duplicate the salient features of *in vitro* burst development [56]. In avalanche analysis, spatiotemporal avalanches are shown to be more close to the original definition of self-organized criticality in [7], and this is proven by presenting its size probability distribution followed the power-law relationship more consistently. In burst analysis, visualization results show that bursts originate at single locations, later develop into a set of active origins, and propagate across the network in a wave-like fashion; their origins and propagation speeds change during network development.

This is the first application of machine learning to spike train data analysis (beyond spike sorting and encoding/decoding) as far as the author is aware. The ML results allow us to conclude that, even in the absence of detailed analysis of pre-burst spiking patterns, there is commonly a localized spatiotemporal pattern of spikes not only providing reliable information about if a burst will occur but also where that burst will start. In addition, our results show that subsequent analysis on the burst trigger patterns can be narrow down to a small number of spikes and neurons, fewer than 50 spikes prior of every burst. For the current investigation, this corresponds to a reduction of more than seven orders of magnitude from the full dataset that contains 600 million spikes.

Chapter 9

FUTURE WORK

First, data from simulations with higher time resolution (e.g. 0.05 ms) may provide a better approximation of mean ISI to group avalanches which in turn identify burst boundaries more accurately, eventually improve our current results.

Second, since fewer than 50 spikes are shown to contain the burst trigger pattern, statistical methods can then be applied to find statistically significant pattens and use them as templates for pattern matching for the entire data. If consistent patterns are found to be predictive of bursts and present before every burst, we can conclude if this is indeed the burst trigger by identifying all occurrence of this particular patten in the data and calculating the relevance between this pattern and bursts.

Last but not least, spatiotemporal spikes were represented as a single train of feature sequences in this work; however, as we visualized bursts using binned images, we found that our data could also be represented by images so that convolutional neural network (CNN) and recurrent neural network (RNN) with long-short term memory (LSTM) can be applied to learn the time-relevant, localized trigger patterns.

BIBLIOGRAPHY

[1] Ö. Gürcan, *Exploration of Biological Neural Wiring Using Self-Organizing Agents.* PhD thesis, Ege University, 09 2013.

[2] S. R. y Cajal, *Recollections of my life*, vol. 8. MIT Press, 1989.

[3] D. A. Wagenaar, J. Pine, and S. M. Potter, "An extremely rich repertoire of bursting patterns during the development of cortical cultures," *BMC neuroscience*, vol. 7, no. 1, p. 11, 2006.

[4] J. van Pelt, P. S. Wolters, M. A. Corner, W. L. Rutten, and G. J. Ramakers, "Long-term characterization of firing dynamics of spontaneous bursts in cultured neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 11, pp. 2051–2062, 2004.

[5] J. Stegenga, J. Le Feber, E. Marani, and W. L. Rutten, "Analysis of cultured neuronal networks using intraburst firing characteristics," *IEEE transactions on biomedical engineering*, vol. 55, no. 4, pp. 1382–1390, 2008.

[6] T. A. Gritsun, J. le Feber, and W. L. Rutten, "Growth dynamics explain the development of spatiotemporal burst activity of young cultured neuronal networks in detail," *PloS one*, vol. 7, no. 9, p. e43352, 2012.

[7] P. Bak, C. Tang, and K. Wiesenfeld, "Self-organized criticality: An explanation of the 1/f noise," *Physical review letters*, vol. 59, no. 4, p. 381, 1987.

[8] J. M. Beggs and D. Plenz, "Neuronal avalanches in neocortical circuits," *Journal of neuroscience*, vol. 23, no. 35, pp. 11167–11177, 2003.

[9] J. M. Beggs and D. Plenz, "Neuronal avalanches are diverse and precise activity patterns that are stable for many hours in cortical slice cultures," *Journal of neuroscience*, vol. 24, no. 22, pp. 5216–5229, 2004.

[10] Y. Yada, T. Mita, A. Sanada, R. Yano, R. Kanzaki, D. J. Bakkum, A. Hierlemann, and H. Takahashi, "Development of neural population activity toward self-organized criticality," *Neuroscience*, vol. 343, pp. 55–65, 2017.

[11] L. Abbott and R. Rohrkemper, “A simple growth model constructs critical avalanche networks,” *Progress in brain research*, vol. 165, pp. 13–19, 2007.

[12] C. Tetzlaff, S. Okujeni, U. Egert, F. Wörgötter, and M. Butz, “Self-organized criticality in developing neuronal networks,” *PLoS computational biology*, vol. 6, no. 12, p. e1001013, 2010.

[13] A. Benucci, R. A. Frazor, and M. Carandini, “Standing waves and traveling waves distinguish two circuits in visual cortex,” *Neuron*, vol. 55, no. 1, pp. 103–117, 2007.

[14] T. K. Sato, I. Nauhaus, and M. Carandini, “Traveling waves in visual cortex,” *Neuron*, vol. 75, no. 2, pp. 218–229, 2012.

[15] R. S. Witte, P. J. Rousche, and D. R. Kipke, “Fast wave propagation in auditory cortex of an awake cat using a chronic microelectrode array,” *Journal of neural engineering*, vol. 4, no. 2, p. 68, 2007.

[16] K. Takahashi, M. Saleh, R. D. Penn, and N. Hatsopoulos, “Propagating waves in human motor cortex,” *Frontiers in human neuroscience*, vol. 5, p. 40, 2011.

[17] Y. Nishitani, C. Hosokawa, Y. Mizuno-Matsumoto, T. Miyoshi, H. Sawai, and S. Tamura, “Variance of spatiotemporal spiking patterns by different stimulated neurons in cultured neuronal networks,” *Int J Academ Res Reflect*, vol. 4, pp. 11–19, 2016.

[18] S. Sakuma, Y. Mizuno-Matsumoto, Y. Nishitani, and S. Tamura, “Simulation of spike wave propagation and two-to-one communication with dynamic time warping,” *AIMS Neuroscience*, vol. 3, no. 4, pp. 474–486, 2016.

[19] C. Thomas Jr, P. Springer, G. Loeb, Y. Berwald-Netter, and L. Okun, “A miniature microelectrode array to monitor the bioelectric activity of cultured cells,” *Experimental cell research*, vol. 74, no. 1, pp. 61–66, 1972.

[20] G. W. Gross, “Simultaneous single unit recording in vitro with a photoetched laser deinsulated gold multimicroelectrode surface,” *IEEE Transactions on Biomedical Engineering*, vol. 26, no. 5, pp. 273–279, 1979.

[21] J. Pine, “Recording action potentials from cultured neurons with extracellular microcircuit electrodes,” *Journal of neuroscience methods*, vol. 2, no. 1, pp. 19–31, 1980.

[22] R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. M. Bower, M. Diesmann, A. Morrison, P. H. Goodman, F. C. Harris, *et al.*, “Simulation of networks of spiking neurons: a review of tools and strategies,” *Journal of computational neuroscience*, vol. 23, no. 3, pp. 349–398, 2007.

[23] E. N. Brown, R. E. Kass, and P. P. Mitra, “Multiple neural spike train data analysis: state-of-the-art and future challenges,” *Nature neuroscience*, vol. 7, no. 5, p. 456, 2004.

[24] T. Tchumatchenko, T. Geisel, M. Volgushev, and F. Wolf, “Signatures of synchrony in pairwise count correlations,” *Frontiers in computational neuroscience*, vol. 4, p. 1, 2010.

[25] G. P. Moore, D. Perkel, and J. Segundo, “Statistical analysis and functional interpretation of neuronal spike data,” *Annual review of physiology*, vol. 28, no. 1, pp. 493–522, 1966.

[26] E. N. Brown, R. Barbieri, U. T. Eden, and L. M. Frank, “Likelihood methods for neural spike train data analysis,” *Computational neuroscience: A comprehensive approach*, pp. 253–286, 2003.

[27] D. R. Brillinger, “Nerve cell spike train data analysis: a progression of technique,” *Journal of the American Statistical Association*, vol. 87, no. 418, pp. 260–271, 1992.

[28] M. Abeles and G. L. Gerstein, “Detecting spatiotemporal firing patterns among simultaneously recorded single neurons,” *Journal of Neurophysiology*, vol. 60, no. 3, pp. 909–924, 1988.

[29] L. Martignon, G. Deco, K. Laskey, M. Diamond, W. Freiwald, and E. Vaadia, “Neural coding: higher-order temporal patterns in the neurostatistics of cell assemblies,” *Neural Computation*, vol. 12, no. 11, pp. 2621–2653, 2000.

[30] D. R. Brillinger, “Maximum likelihood analysis of spike trains of interacting nerve cells,” *Biological cybernetics*, vol. 59, no. 3, pp. 189–200, 1988.

[31] E. Chornoboy, L. Schramm, and A. Karr, “Maximum likelihood identification of neural point process systems,” *Biological cybernetics*, vol. 59, no. 4-5, pp. 265–275, 1988.

[32] E. Schneidman, M. J. Berry II, R. Segev, and W. Bialek, “Weak pairwise correlations imply strongly correlated network states in a neural population,” *Nature*, vol. 440, no. 7087, p. 1007, 2006.

[33] J. Shlens, G. D. Field, J. L. Gauthier, M. I. Grivich, D. Petrusca, A. Sher, A. M. Litke, and E. Chichilnisky, “The structure of multi-neuron firing patterns in primate retina,” *Journal of Neuroscience*, vol. 26, no. 32, pp. 8254–8266, 2006.

[34] A. Tang, D. Jackson, J. Hobbs, W. Chen, J. L. Smith, H. Patel, A. Prieto, D. Petrusca, M. I. Grivich, A. Sher, *et al.*, “A maximum entropy model applied to spatial and temporal correlations from cortical networks in vitro,” *Journal of Neuroscience*, vol. 28, no. 2, pp. 505–518, 2008.

[35] M. Canepari, M. Bove, E. Maeda, M. Cappello, and A. Kawana, “Experimental analysis of neuronal dynamics in cultured cortical networks and transitions between different patterns of activity,” *Biological cybernetics*, vol. 77, no. 2, pp. 153–162, 1997.

[36] I. H. Stevenson and K. P. Kording, “How advances in neural recording affect data analysis,” *Nature Neuroscience*, vol. 14, pp. 139 EP –, 01 2011.

[37] J. M. Nageswaran, N. Dutt, J. L. Krichmar, A. Nicolau, and A. V. Veidenbaum, “A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors,” *Neural networks*, vol. 22, no. 5-6, pp. 791–800, 2009.

[38] R. Brette and D. F. Goodman, “Simulating spiking neural networks on gpu,” *Network: Computation in Neural Systems*, vol. 23, no. 4, pp. 167–182, 2012.

[39] M. Stiber, F. Kawasaki, D. B. Davis, H. U. Asuncion, J. Y.-H. Lee, and D. Boyer, “Braingrid+ workbench: High-performance/high-quality neural simulation,” in *Neural Networks (IJCNN), 2017 International Joint Conference on*, pp. 2469–2476, IEEE, 2017.

[40] D. L. Hudson and M. E. Cohen, *Neural networks and artificial intelligence for biomedical engineering.* Wiley Online Library, 2000.

[41] I. Inza, B. Calvo, R. Armañanzas, E. Bengoetxea, P. Larrañaga, and J. A. Lozano, “Machine learning: an indispensable tool in bioinformatics,” in *Bioinformatics methods in clinical research*, pp. 25–48, Springer, 2010.

[42] J. W. McClurkin, L. M. Optican, B. J. Richmond, and T. J. Gawne, “Concurrent processing and complexity of temporally encoded neuronal messages in visual perception,” *Science*, vol. 253, no. 5020, pp. 675–677, 1991.

[43] T. W. Kjaer, J. A. Hertz, and B. J. Richmond, “Decoding cortical neuronal signals: network models, information estimation and spatial tuning,” *Journal of computational neuroscience*, vol. 1, no. 1-2, pp. 109–139, 1994.

[44] J. K. Chapin and M. A. Nicolelis, “Principal component analysis of neuronal ensemble activity reveals multidimensional somatosensory representations,” *Journal of neuroscience methods*, vol. 94, no. 1, pp. 121–140, 1999.

[45] B. R. Cowley, M. T. Kaufman, Z. S. Butler, M. M. Churchland, S. I. Ryu, K. V. Shenoy, and M. Y. Byron, “Datahigh: graphical user interface for visualizing and interacting with high-dimensional neural activity,” *Journal of neural engineering*, vol. 10, no. 6, p. 066012, 2013.

[46] D. M. Endres, P. Földiák, and U. Priss, "An application of formal concept analysis to semantic neural decoding," *Annals of Mathematics and Artificial Intelligence*, vol. 57, no. 3-4, pp. 233–248, 2009.

[47] D. Picado-Muiño, C. Borgelt, D. Berger, G. L. Gerstein, and S. Grün, "Finding neural assemblies with frequent item set mining," *Frontiers in neuroinformatics*, vol. 7, p. 9, 2013.

[48] P. Quaglio, A. Yegenoglu, E. Torre, D. M. Endres, and S. Grün, "Detection and evaluation of spatio-temporal spike patterns in massively parallel spike train data with spade," *Frontiers in computational neuroscience*, vol. 11, p. 41, 2017.

[49] J. W. Pillow, Y. Ahmadian, and L. Paninski, "Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains," *Neural computation*, vol. 23, no. 1, pp. 1–45, 2011.

[50] E. Astrand, P. Enel, G. Ibos, P. F. Dominey, P. Baraduc, and S. B. Hamed, "Comparison of classifiers for decoding sensory and cognitive information from prefrontal neuronal populations," *PloS one*, vol. 9, no. 1, p. e86314, 2014.

[51] J. I. Glaser, R. H. Chowdhury, M. G. Perich, L. E. Miller, and K. P. Kording, "Machine learning for neural decoding," *arXiv preprint arXiv:1708.00909*, 2017.

[52] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Computation in Neural Systems*, vol. 9, no. 4, pp. R53–R78, 1998.

[53] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural computation*, vol. 16, no. 8, pp. 1661–1687, 2004.

[54] R. J. Vogelstein, K. Murari, P. H. Thakur, C. Diehl, S. Chakrabartty, and G. Cauwenberghs, "Spike sorting with support vector machines," in *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, vol. 1, pp. 546–549, IEEE, 2004.

[55] P. Horton, A. Nicol, K. Kendrick, and J. Feng, "Spike sorting based upon machine learning algorithms (soma)," *Journal of neuroscience methods*, vol. 160, no. 1, pp. 52–68, 2007.

[56] F. Kawasaki and M. Stiber, "A simple model of cortical culture growth: burst property dependence on network composition and activity," *Biological cybernetics*, vol. 108, no. 4, pp. 423–443, 2014.

[57] L. F. Abbott, "Lapicque's introduction of the integrate-and-fire model neuron (1907)," *Brain research bulletin*, vol. 50, no. 5-6, pp. 303–304, 1999.

[58] J. E. Lisman, "Bursts as a unit of neural information: making unreliable synapses reliable," *Trends in neurosciences*, vol. 20, no. 1, pp. 38–43, 1997.

[59] E. M. Izhikevich, N. S. Desai, E. C. Walcott, and F. C. Hoppensteadt, "Bursts as a unit of neural information: selective communication via resonance," *Trends in neurosciences*, vol. 26, no. 3, pp. 161–167, 2003.

[60] D. Plenz and T. C. Thiagarajan, "The organizing principles of neuronal avalanches: cell assemblies in the cortex?," *Trends in neurosciences*, vol. 30, no. 3, pp. 101–110, 2007.

[61] E. Mjolsness and D. DeCoste, "Machine learning for science: state of the art and future prospects," *science*, vol. 293, no. 5537, pp. 2051–2055, 2001.

[62] E. Alpaydin, *Introduction to machine learning.* MIT press, 2014.

[63] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach.* Malaysia; Pearson Education Limited,, 2016.

[64] E. Maeda, H. Robinson, and A. Kawana, "The mechanisms of generation and propagation of synchronized bursting in developing networks of cortical neurons," *Journal of Neuroscience*, vol. 15, no. 10, pp. 6834–6845, 1995.

[65] P. Bak, C. Tang, and K. Wiesenfeld, "Self-organized criticality," *Physical review A*, vol. 38, no. 1, p. 364, 1988.

[66] J. Eichhorn, A. Tolias, A. Zien, M. Kuss, J. Weston, N. Logothetis, B. Schölkopf, and C. E. Rasmussen, "Prediction on spike data using kernel algorithms," in *Advances in neural information processing systems*, pp. 1367–1374, 2004.

[67] M. Chiappalone, A. Novellino, I. Vajda, A. Vato, S. Martinoia, and J. van Pelt, "Burst detection algorithms for the analysis of spatio-temporal patterns in cortical networks of neurons," *Neurocomputing*, vol. 65, pp. 653–662, 2005.

[68] S. Grün and S. Rotter, *Analysis of parallel spike trains*, vol. 7. Springer, 2010.

[69] Y. Mukai, T. Shiina, and Y. Jimbo, "Continuous monitoring of developmental activity changes in cultured cortical networks," *Electrical Engineering in Japan*, vol. 145, no. 4, pp. 28–37, 2003.

[70] R. Segev, M. Benveniste, E. Hulata, N. Cohen, A. Palevski, E. Kapon, Y. Shapira, and E. Ben-Jacob, “Long term behavior of lithographically prepared in vitro neuronal networks,” *Physical review letters*, vol. 88, no. 11, p. 118102, 2002.

[71] K. Takahashi, S. Kim, T. P. Coleman, K. A. Brown, A. J. Suminski, M. D. Best, and N. G. Hatsopoulos, “Large-scale spatiotemporal spike patterning consistent with wave propagation in motor cortex,” *Nature communications*, vol. 6, p. 7169, 2015.