

# Yakın Kaynak Deprem Kayıtlarındaki Hız Atımlarının Makine Öğrenimi Teknikleri ile Belirlenmesi

Deniz Ertuncay, Giovanni Costa

Trieste Üniversitesi, Matematik ve Yer Bilimleri Bölümü, Trieste, İtalya

deniz.ertuncay@phd.units.it, costa@units.it



## Veri

Bu çalışmada görece büyük ( $M_w \geq 6.0$ ) ve kabukta meydana gelmiş (odak derinliği  $< 100$  km) depremlerin yakın kaynak deprem kayıtları (deprem-istasyon uzaklığı  $< 100$  km) kullanılmıştır. Kullandığımız veri tabanı sayesinde dünyadaki bir çok aktif, deprem üreten ve sismik risk yaratan bölge kapsamıştır (Şekil 1). Çeşitli odak mekanizmalarına sahip 200'den fazla deprem için 19.000'den fazla deprem kaydı kullanılmıştır.

Hız atımları hız kayıtlarında ivme kayıtlarına göre daha kolay belirlenebildiğinden [2] [3] dolayı bu çalışmada hız kayıtları kullanılmıştır. Deprem kayıtlarının her istasyondaki 3 farklı birleşeninin hız kayıtları ve spektrogram kayıtları [5] kullanılmıştır (Şekil 2).

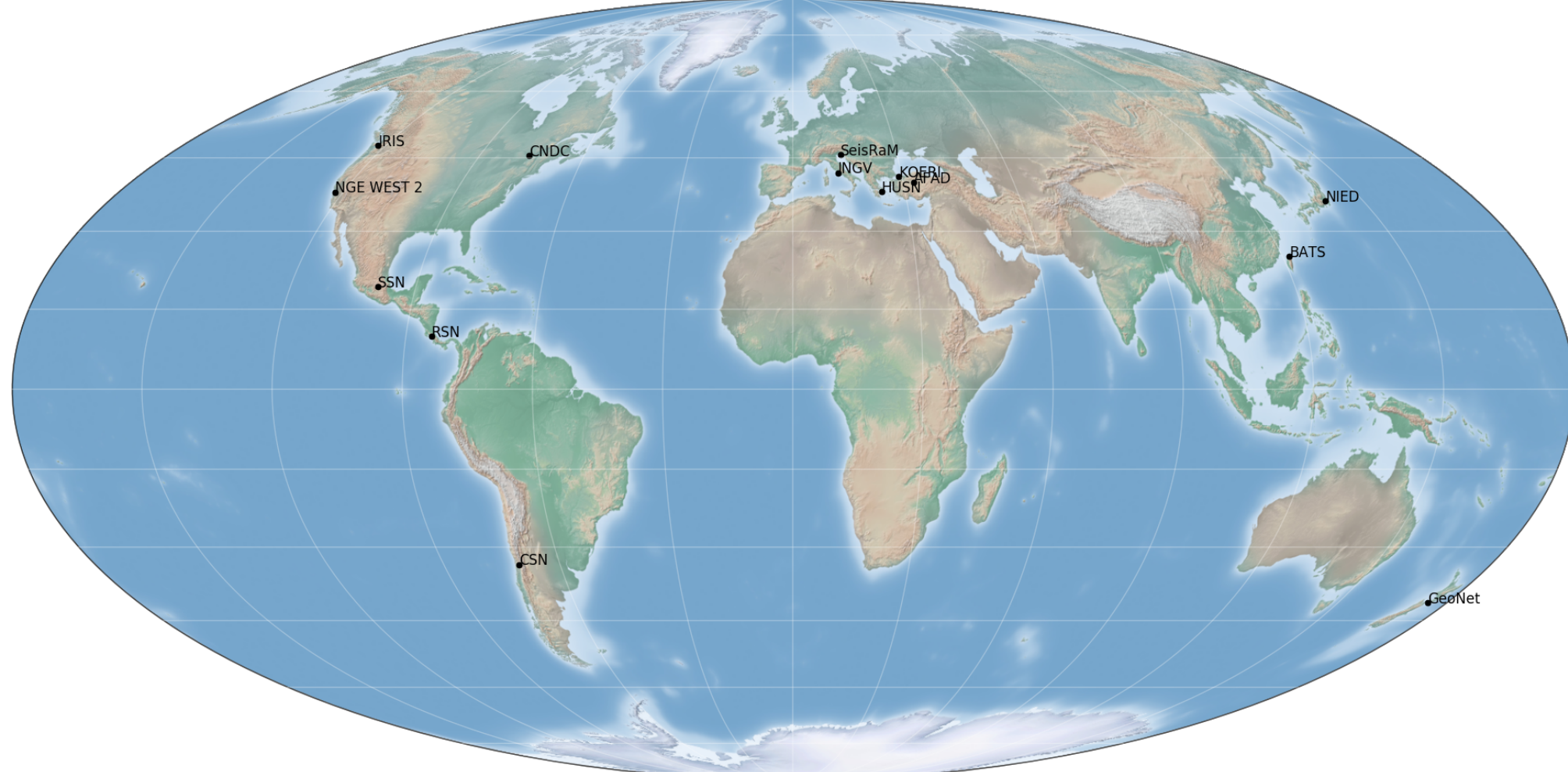


Figure 1: Veritabanımızdaki sismik ağlar.

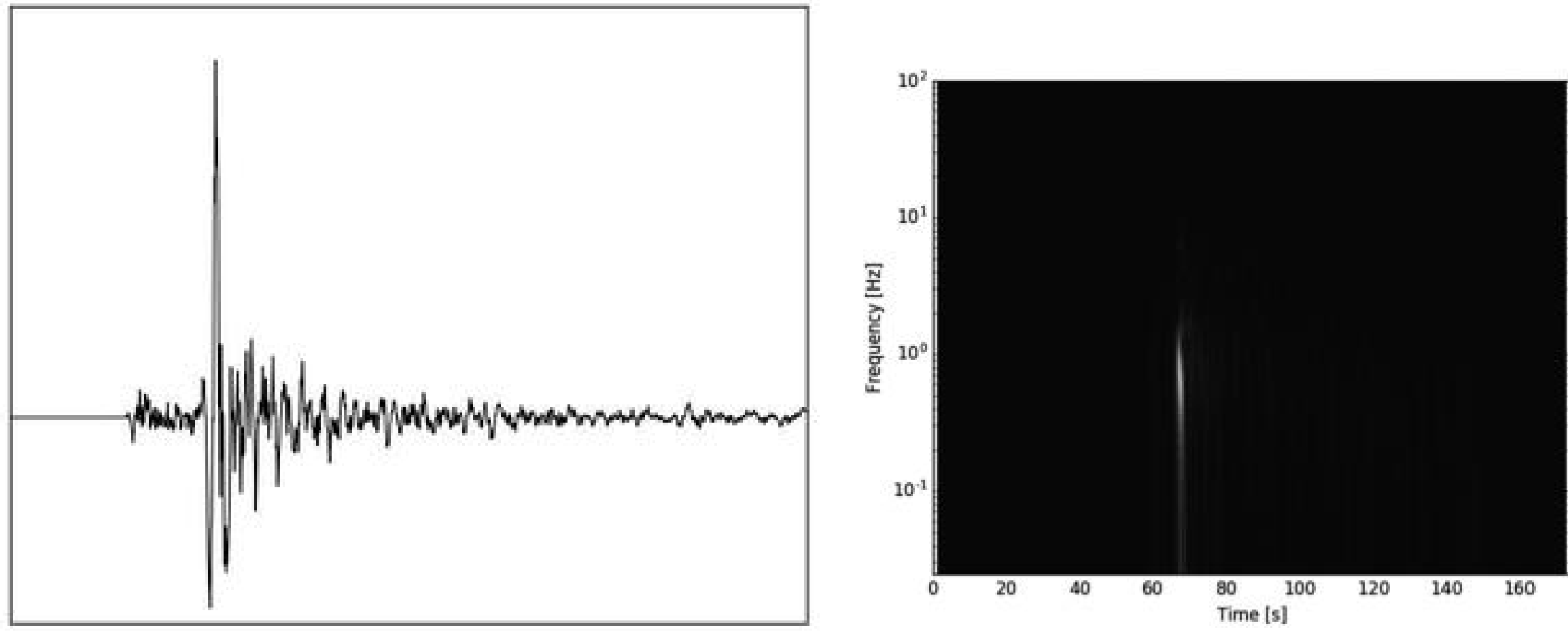


Figure 2: 5 Ekim 2015'de meydana gelmiş,  $M_w$  6.1 büyüklüğündeki depremin Şili Üniversitesi, Ulusal Sismoloji Merkezinin C100 adlı istasyonunun kaydettiği deprem kaydının hız ve spektrogram kayıtları.

## Metod

### Hız Atımı Sinyalinin Belirlenmesi

Fayın sismik istasyon yönünde kırılması veya fayın kırılma hızının ortamın kesme dalgası hızına yakın olması, sismik enerjinin deprem kayıtcısına deprem kaydının başında tek ve geniş bir atım olarak ulaşmasına neden olur. Bu atım hareketi faydan gelen sismik yayılım enerjinin kümülatif toplamının faya neredeyse tek bir anda gelmesine neden olur. Faydaki kesme dalgası yer değiştirmesinin yayılım örüntüsü bu büyük atıma neden olur. Bu atım faya dik olarak konumlanmış (veya döndürülmüş) deprem istasyonlarında doğrultuya dik yöndeki bileşende doğrultuya paralel yöndeki bileşene göre daha net görülür.

### Baker'ın Yöntemi

Baker [2] atım gösteren kayıtları atım göstermeyen kayıtlardan ayırmak için dalgacık (wavelet) analizini kullanmıştır. En büyük hız atımı dalgacık ayrıştırma (wavelet decomposition) kullanılarak yapılmıştır. 4. dereceden Daubechies dalgacığı ana dalgacık olarak kullanılmıştır. Bunun nedeni ise Daubechies dalgacığının hız atımlarını taklit etmedeki başarısıdır.

### Chang'in Yöntemi

Chang [3] bütün deprem kaydı ile deprem kaydındaki en büyük hız değerinin olduğu alanın (PGV) enerji oranlarını karşılaştırmıştır. PGV değerinin 30 cm/s'den daha büyük olması şartı veri analizinde kullanılmıştır (Bu şart Baker'ın yönteminde de kullanılmıştır). Hız atım modeli aşağıdaki gibi oluşturulmuştur:

$$v_p(t; A_p, T_p, N_c, T_{pk}, \theta) = A_p \exp\left[-\frac{\pi^2}{4} \left(\frac{t - T_{pk}}{N_c T_p}\right)^2\right] \times \cos\left(2\pi \frac{t - T_{pk}}{T_p} - \theta\right) \quad (1)$$

Burada  $v_p$  çıkartılan atımı,  $t$  zaman serisini,  $A_p$  çıkartılan hız atımının genliğini,  $T_p$  hız atımının periyodunu,  $N_c$  hız atımındaki devir sayısını,  $T_{pk}$  hız atımının zaman verisindeki yerini ve  $\theta$  hız atımının fazını temsil eder. Eğer ki PGV'nin olduğu bölgenin enerjisinin oranı bütün deprem kaydının enerjisinin oranının 0.3'üne eşit veya bu değerden büyükse, bu sismik kayıt hız atımı özelliği gösterir deniz. Bu yöntemin matematiksel gösterimi aşağıdaki gibidir:

$$E_p = \frac{\int_{t_s}^{t_e} v^2(t) dx}{\int_0^\infty v^2(t) dx} \quad (2)$$

$t_s$  ve  $t_e$  hız atımının başlangıç ve bitiş anlarının zaman serisi üzerindeki yerini temsil ederler.

## Makine Öğrenimi Algoritmaları

Scikitlearn [1], Keras [4] ve Tensorflow [6] verinin işlenmesi için kullanılmıştır. Metodlar evrişimli sinir ağları ve darboğaz bölümlerinde açıklanmıştır. Veri akış şemaları Şekil 3'de görülebilir. Sınıflandırma işleminde imaj verisi kullanılmıştır.

### Evrişimli Sinir Ağları (CNN)

**Model 1** Her katmanda 2B evrişim, ilk iki katmanda 32 ve üçüncü katmanda 64 filtre kullanılmıştır. Daha sonra havuzlama ve tam bağlantılı katman 128 sinir ağıyla kullanılmıştır. Bu işlemin sonuçları Tablo ??'deki Model 1'de görülebilir.

**Model 2** Her katmanda 2B evrişim  $3 \times 3$  çekirdek (kernel) boyutu, 32 filtre ve ReLu aktifleştirme fonksiyonu kullanılmıştır. Yoğun fonksiyonlar sırasıyla 128 ve 1 birim ve ReLu ve sigmoid aktivasyon fonksiyonları kullanmıştır. Bu işlemin sonuçları Tablo ??'deki Model 2'de görülebilir.

### Darboğaz (Bottleneck)

Tensorflow'un darboğaz yöntemini kendi veritabanımıza uyguladık. İlk aşamada veri tabanındaki bütün veriler analiz edildi ve her biri için darboğaz değeri bulundu. Darboğaz terimi sınıflandırma yapılmadan hemen önceki katmanı tanımlamak için oluşturulmuş bir terimdir. Bu katman sınıflandırılması istenen sınıfları tanımlamak için eğitilen bir katmandır. Bu demektir ki bu katmanın yaptığı işlem bütün imajı tanımlamak için yapılan işlemlerdeki bilgiyi doğru bir şekilde ve az kaynak harcayarak yapabilmektedir. Bu işlemin sonucu Tablo 1'deki Model 3'de görülebilir.

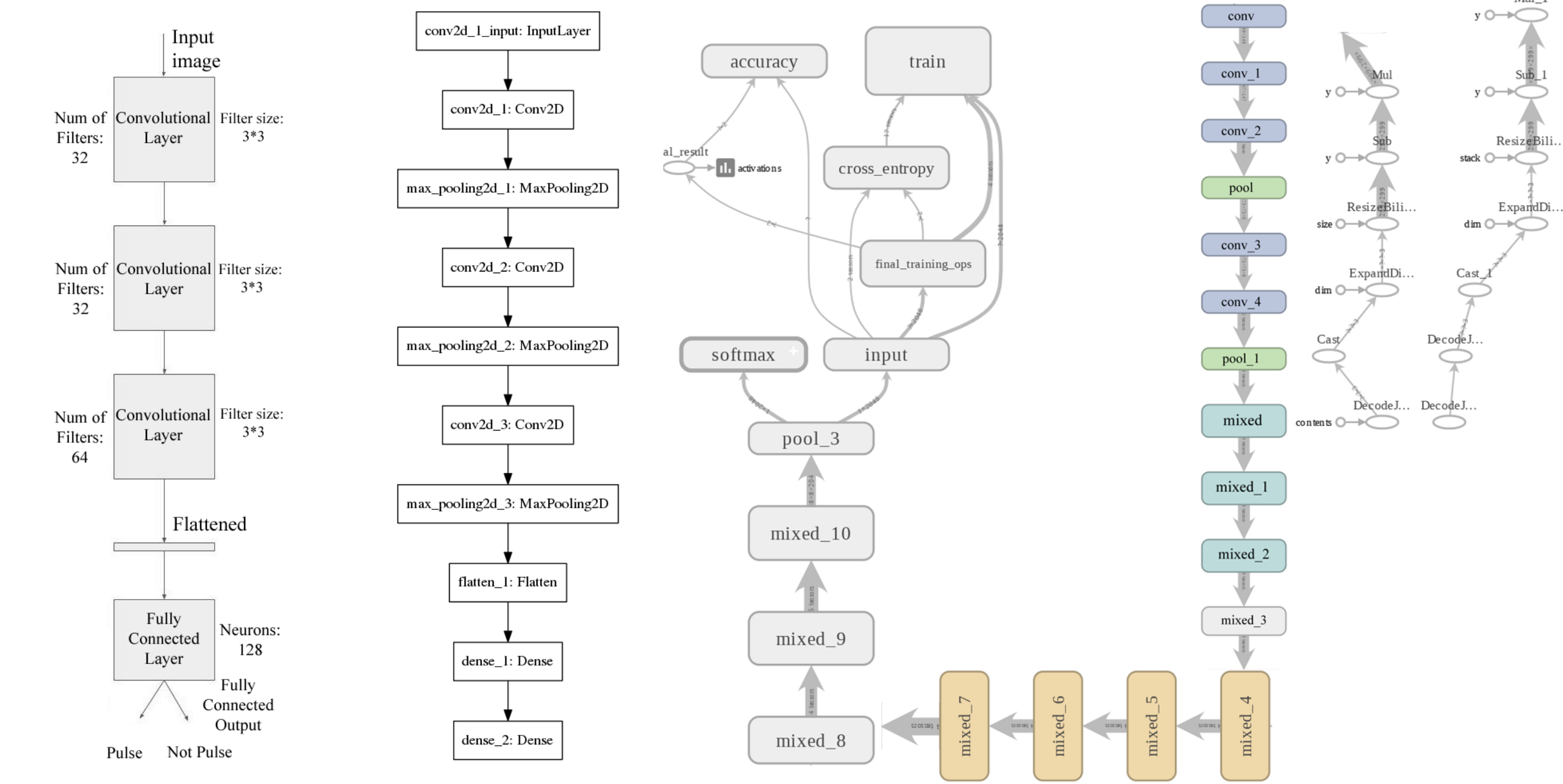


Figure 3: Bütün modellerin veri akış şemaları. Soldan sağa sırasıyla Model 1, Model 2 ve Model 3.

## Sonuçlar

		Model 1			Model 2			Model 3
		Hız Atımı Değil (%)	Hız Atımı (%)	Doğ. (%)	Hız Atımı Değil (%)	Hız Atımı (%)	Doğ. (%)	Doğ. (%)
Baker 2007	Deprem Dalgası	97.83	43.48	71.60	99.95	2.19	97.58	84.4
	Spektrogram	100	0	50	99.22	43.80	97.86	83.9
Chang 2016	Deprem Dalgası	99.24	12.12	55.68	100	4.38	97.69	82.6
	Spektrogram	100	0	50	98.66	43.07	97.30	83.4

Table 1: Farklı modellerin sonuçları. Hız Atımı Değil ve Hız Atımı yüzdeleri doğru sonuçların yüzdelidir. Doğruluk (Doğ.) ise genel doğruluk oranını gösterir.

### Eksik Yönler

- Yüksek miktarda verinin işlenmesi gerekiyor.
- Deprem kayıtlarını imaj olarak işlemek etkili değil.
- Modelin eğitilmesi için çok fazla sayıda deprem dalgası gerekiyor.

## Sonuçlar

- Hız atımlarının belirlenmesinde imaj verisi üzerinden çalışan makine öğrenimi algoritmaları iyi bir çözüm sunmuyor.
- Katmanlar iyi bir şekilde tanımlanabilirse imaj tanınması yöntemi işe yarayabilir.
- Katman mimarisine bağlı olarak deprem dalgası ve spektrogramlar daha verimli analiz edilebilir.
- Model 1 deprem verisini sınıflandırmada Model 2'ye göre daha başarılı bir sonuç vermiştir. Model 2 ise spektrogram sınıflandırmasına Model 1'e göre daha iyi bir sonuç vermiştir.
- Model 3 deprem verisi ve spektrogram verisinin sınıflandırmasında en iyi sonucu vermiştir.

## Gelecek Planları

- Veriyi zamanın bir fonksiyonu olarak ele almak.
- Daha basit makine öğrenimi işlemleri yapabilmek için deprem verisine ait yüksek seviye bilgiler belirlenmesi.
- Bu yüksek seviye bilgilerin mühendislik sismolojisinde yoğun olarak kullanılan başka parametrelerle birleştirerek devamlı zaman verisi üzerinden bir sınıflandırma algoritması geliştirilmesi.

## Teşekkür

Ulusal Oşinografi ve Deneysel Jeofizik Enstitüsünden (OGS) Dr. Stefano Parolai'ye tavsiyeleri, Güneybatı Jiaotong Üniversitesi İnşaat Mühendisliği Okulu Bölümünden Yard. Doç. Dr. Zhiwang Chang'ye kendi metodunun kodunu bizimle paylaştığı için ve İleri Düzey Çalışmalar için Uluslararası Okul (SISSA), Yoğun Madde Fiziği Bölümünden Dr. Emine Küçükbenli'ye makine öğrenimi algoritmaları ile ilgili tavsiyeleri için teşekkür ederiz.

## References

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.
- [2] Jack W Baker. Quantitative classification of near-fault ground motions using wavelet analysis. *Bulletin of the Seismological Society of America*, 97(5):1486–1501, 2007.
- [3] Zhiwang Chang, Xiaodan Sun, Changhai Zhai, John X Zhao, and Lili Xie. An improved energy-based approach for selecting pulse-like ground motions. *Earthquake Engineering & Structural Dynamics*, 45(14):2405–2411, 2016.
- [4] François Chollet et al. Keras. <https://github.com/keras-team/keras>, 2015. Accessed: 2018-02-28.
- [5] Banu Mena and P Martin Mai. Selection and quantification of near-fault velocity pulses owing to source directivity. *Georisk*, 5(1):25–43, 2011.
- [6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.